Machine learning coursework

UNIVERSITY
OF SUSSEX

Andres Sebastián Salazar Alturo

Candidate number: 276209

Lecturer:
Dr Temitayo Olugbade

Module:

Machine Learning

University of Sussex
Artificial Intelligence and adaptive systems MSc
Brighton, UK
2024

Performance:

The model was evaluated with multiple techniques. Each metric offers unique perspective on the model’s performance, highlighting different aspects of the behavior and prediction quality of the model. The metrics used were Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-Squared and Mean Squared Error (MSE).

Mean Absolute Error (MAE): The MAE is the average absolute difference between the predicted values and the actual values. The fact that the MAE is approximately equal to 428,901.78 means that, on average, the model's predictions are off by this amount in comparison to the actuals. A higher value indicates significant errors in numerous predictions. The magnitude of this error should be judged compared to the scale of the target variable (export values). These values have significant variations, some of them are hundreds, thousands, and millions. However, when the actual value is smaller, the model does not adequately predict those values.

MAE mathematical expression:

$$MAE = \frac{1}{n}\sum_{I=1}^{n} |y_i - \hat{y}_i|$$

Where:

- y_i is the actual value
- $\hat{y}_i$ is the predicted value
- n is the number of observations

Root Mean Squared Error (RMSE): RMSE can be helpful to identify when large errors exist, particularly in the presence of high sensitivity to large errors that can distort the average. Given that there are big differences in the export value by country, RMSE offers a different perspective on the model performance. RMSE takes the average of the squares of the residuals, thus, it is highly affected by large errors, which are substantial. A value of 1,653,558.38 is on the higher side and might suggest that there is a substantial difference between some of the predicted values and the true export values. This metric, therefore, indicates that the model could be catching up on the general trend but not predicting more accurate values, most likely because of the effects of outliers or underfitting the model due to the strategy used to fill the null values, a better strategy to impute the values may be beneficial.

RMSE mathematical expression:

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

Where:

- y_i is the actual value
- $\hat{y}_i$ is the predicted value
- n is the number of observations

R-Squared: It is a good metric for comparison of the quality of fitting between various regression models. The closer the value of R-squared to 1.0, the better the explanatory power of the model. The R-squared of 0.694 shows that approximately 69.4% of the variance in the dependent variable (export value) is explained by the independent variables (features). This indeed lies at the scale of moderate predictive accuracy, yet it leaves a considerable part of the variation in the residual.

R-Squared mathematical expression:

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

Where:

- $\bar{y}$ is the mean of the actual values y
- y_i is the actual value
- $\widehat{y_i}$ is the predicted value

The dataset built based on the given data had a total of 10,147 instances. For the purposes of the experimentation and predict the export value in the years 2018 to 2022. The dataset was split into two subsets: the training set and the test set. The training set consisted in 8,744 instances that corresponded to the data from 1980 to 2017. The remaining 1,184 instances represented the data from 2018 to 2022. The testing data was exclusively used for assessing the model's performance on unseen data. Then, the data was scaled to be used in the Multi-Layer Perceptron model.

MLP regressor model:

Multi-Layer Perceptron, being powerful, can model very complex. It acts as universal approximator to any continuous function theoretically, in very high dimensions. Its versatility and implementation of libraries on a vast scale in different and dynamic data environments make them very powerful.

The architecture used in the proposed MLP solution was:

Hidden layers and neurons: The model had two hidden layers, the first with 100 neurons and the second with 50 neurons. It was decided to use just two hidden layers to not increase the complexity of the model and reduce the computational complexity. Moreover, different numbers of neurons were used to assess the performance of the model. However, the selected configuration had a good performance compared to the other hidden layers and neurons configurations.

Activation function: The activation function used for the hidden layers was the Rectified Linear Unit (ReLU). It is computationally more efficient compared to other activation functions like sigmoid or tanh, as it involves only simpler mathematical operations like the maximum of zero and the input value. Furthermore, introduces nonlinearity into the network, even though it appears piecewise linear, which helps in approximating complex functions and the relationships between variables with a deep neural network. Also, helps in avoiding the vanishing gradient problem that arises quite often in the use of other activation functions. It does this because the gradient is a constant function and equals 1 for all positive inputs, so the gradient does not vanish in backpropagation.

ReLU mathematical expression:

$$f(x) = \max(0, x)$$

For positive inputs, ReLU output is just the value passed. This direct mapping for positive values also keeps gradients large and consistent during training, so the common problem of the gradient vanishing too much through backpropagation is not encountered in deep networks with other activation functions. With inputs that are negative, the ReLU outputs zero, which makes the network sparse but, at the same time, helps in faster computation during forward and backward passes.

Loss function: MSE is the average of squared errors, it is the difference between the actual value and the estimated value. It gives an intuitive way to measure the error in the model. Gives larger weight to bigger differences. Large errors weigh in their contribution to the total MSE, which also makes the model especially sensitive to large errors. The role of the MSE in training is that is the reference value that the MLP model learns to reduce the loss of MSE, meaning that the weights are changed in such a way that the error between the true predicted output and the actual output from the training data is at a minimum. The method through which this is achieved is backpropagation: the error is propagated back through the network to update the weights.

MSE mathematical expression:

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

Where:

- n is the number of samples
- y_i is the actual value of the sample
- $\widehat{y_i}$ is the predicted value

For each prediction is calculated the difference between the predicted value and the actual value $(y_i - \widehat{y_i})$. Each difference is squared. Squaring is used because the negative signs will be removed (Negative and positive errors are equally bad), and higher errors will be penalized more than the smaller ones. Finally, results are averaged in a single measure of the prediction performance across all the observations. This average represents the expected value of the squared differences, giving a comprehensive view of model accuracy.

Number of units in the output layer: The model is set to only have one output layer, which is what is needed when the task is to perform a regression analysis, where the target is one single outcome to be predicted, that in this case, is the export value. This layer is linearly activated and, performs as an approximate linear weighting sum of the inputs coming from the last hidden layer, adjusted by a bias term. This setup allows for quantitative predictions to be very precise, and, therefore, this model is useful in this regression task.

Overfitting prevention: To prevent overfitting in the model. Some parameters were used in the MLP regressor instance. The techniques used were:

- L2 regularization
- Early stopping
- Reducing model complexity

L2 regularization:

$$L = L_0 + \lambda \sum_{i=1}^{n} {w_i}^2$$

Where:

- L_0, original loss function
- λ is the regularization parameter that controls the strength of the regularization penalty
- w_i are the weights of the model
- n is the total number of weights in the model

L2 regularization tries to keep the model weights (that is, the coefficients) low and therefore the model simple. Lower weights allow the model to be less sensitive, meaning that small changes in input features will have a more modest impact on the output. This reduced sensitivity, helps with generalization to unseen data and does not help to memorize the noise or small changes in the data. Without regularization, the sole objective is to minimize this error, potentially leading to a complex model that fits the training data very closely but performs poorly on new, unseen data.

Early stopping: Is a regularization method applied to avoid overfitting the training of a learning algorithm by iteration. It is based on the monitoring of the performance of the model with a validation set across training time. If a performance does not improve beyond the value of a variable called 'patience' over a certain number of epochs, the training process will stop, and the weights will be rolled back to where they gave the best performance on the validation set.

Reducing model complexity: Simplification of the model means changing the architecture of the model to reduce the number of parameters, which in this case was used fewer hidden layers or fewer neurons per layer. This way, model simplification prevents overfitting because a less complex model has less ability to learn detailed noise within the training data, thereby promoting generalization to unseen data.

Features and labels:

Initially there were 13 different data sets that covered a category of variables relevant to food and agriculture. The data was provided by Dr Temitayo Olugbade. The data was extracted from FAOSTAT database, which gives open access to food and agricultural data for over 245 countries and covers from the mid-1990s to present day.

The first step was to decide relevant data to predict the export value of a country effectively. Due to the extension of the dataset, it was necessary to compare the information available for each country. Meaning that in some cases, countries had information from 1990 to 2023 and other countries had information from 2000 to 2023. This was a recurrent pattern in the dataset, that in some cases when merging the data, multiple empty values appeared for years that there was no available information, and some countries had more information than other, making difficult to select properly the features.

After scanning all the available data and understanding deeply the available data, the following variables were selected as potentially relevant features to predict the export value.

- Food trade indicators
  - Export value
  - Import value
- Consumer price indicators
  - Food inflation
- Food balances
  - Export quantity
  - Import quantity
  - Losses for crop and live stock products
- Exchange rate
  - Local currency units per USD
- Land use
  - Agricultural land
  - Permanent crops
- Food security
  - Political stability and absence of violence/terrorism (index)
- Foreign direct investment (FDI)
  - Total FDI inflow
  - Total FDI outflow

Each variable was extracted from the original dataset and exported as csv file. The design decision made at this point was to handle zero values presented with the average value of the area in that year. E.g., If there was a zero value that was identified as outlier or not relevant. That value was replaced by the average value of the country in that year. Then, the duplicate values were removed, hence, the variables had a value for the whole year, and it was kept as much information as possible.

Afterwards, the files were merged into one csv file. To merge the variables the columns ‘Area’ and ‘Year’ were the common columns for the datasets. The final data set had the area and year and the variables as columns. The last column was the export value, that was the column used as target for the Multi-Layer Perceptron model. The final data set had multiple NaN values; these values were replaced with zeros. Subsequently, the zeros were replaced with the median value. The median imputation helped to preserve the distribution of the data, more with the data used that was skewed and had outliers. The method provided a more robust measure of the central tendency than the mean and ensures that the data retains its integrity for subsequent analysis and machine learning modeling.

Preprocessing:

Before training the model, the dataset was scaled. This is important because the model is based on numerical optimization, that is the optimization used in neural networks, which depends mainly on the optimization algorithm, such as gradient descent. StandardScaler scales features in a range where every feature has a mean of 0 and unit variance. Optimize the efficiency of the gradient descent, and faster convergence is obtained, since the features are normalized, enabling equal contribution from each feature. This ensures that more weight is not given to features having larger numeric ranges, thus keeping every feature important in the model. It avoids numerical instability that is common in problems which are concerned with the neural network if the input scales are not appropriate. Therefore, StandardScaler was used for training an effective and stable model and is also used to make more reliable predictions.

The library used to scale the dataset was scikit-learn that has a class named StandardScaler that performs the operation. The transformation is based on the following mathematical equations:

Mean removal:

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

Where:

- x_i represents individual samples, and n is the total number of samples.

Standard deviation scaling:

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

Then, scale the data:

$$x_i' = \frac{x_i - \mu}{\sigma}$$

Where:

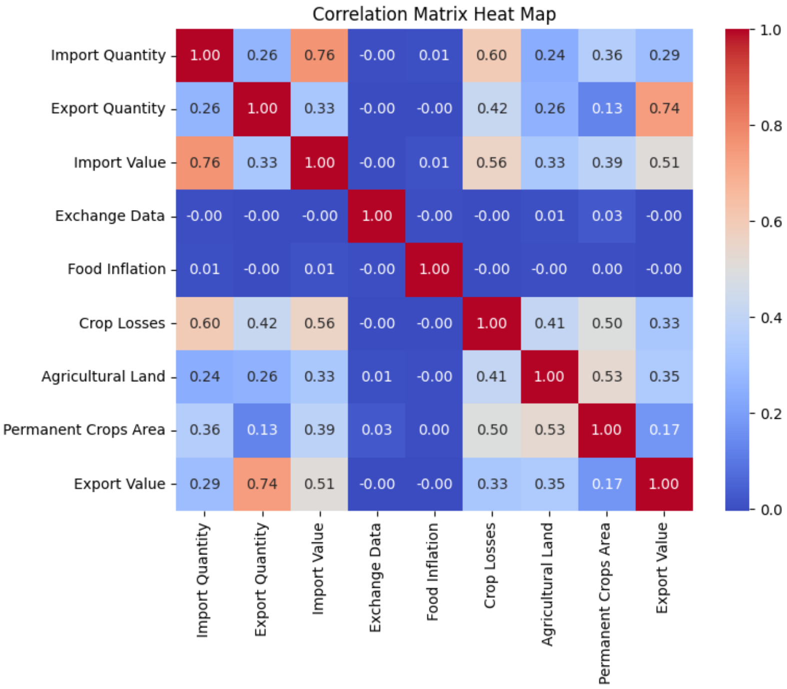
- x_i' is the scaled value

Finally, to complete the preprocessing the correlation matrix heat map was created. From the map is possible to note the correlation between the features and the target.

- Import Quantity:
    - Export Value: Correlation of 0.293 suggests a moderate positive association. Higher import quantities might indicate a robust trade environment, possibly boosting export values.
- Export Quantity:
    - Export Value: Strong correlation of 0.739, indicating that the volume of goods exported strongly influences the monetary value of exports.
- Import Value:
    - Export Value: Moderate correlation of 0.506 implies that higher import values are often related to higher export values, perhaps indicative of active trade sectors in higher GDP countries.
- Exchange Data:
    - Export Value: Very weak correlation of -0.0026, indicating nearly no linear relationship between exchange rate data and export value. Exchange rates might not directly impact on the export value when other economic factors are considered.
- Food Inflation:
    - Export Value: Almost no correlation of -0.0024 suggests that general food inflation rates do not significantly affect the value of exports directly.
- Crop Losses:
    - Export Value: Moderate correlation of 0.330 indicates that areas with significant crop losses might see higher export values, perhaps due to price increases of scarce food.
- Agricultural Land:
    - Export Value: Moderate correlation of 0.345 suggests that larger areas dedicated to agriculture might support higher export values, potentially through increased agricultural outcome.
- Permanent Crops Area:
    - Export Value: Lower correlation of 0.174 suggests a smaller, yet positive relationship, indicating that the area under permanent crops plays a less significant role in influencing export values.
- Political Stability Index:
    - Export Value: Weak correlation of 0.061 indicates that political stability has a marginal positive impact on export value. More stable political environments might foster better economic conditions, slightly increasing export values.
- Total FDI inflow:
    - Export Value: Moderate correlation of 0.435 shows that higher FDI inflows are associated with higher export values, potentially reflecting better economic conditions and investment climates that favor export growth.

- Total FDI outflow:
    - Export Value: Moderate correlation of 0.554, stronger than FDI inflow, suggesting that countries that invest more abroad may also have higher export values, possibly due to multinational operations and global trade integration.

After the analysis of the correlation matrix and the relation with the export value. It was decided that exchange data and food inflation were not relevant features for the model. Using them would not be beneficial to the model and may cause problems with the predictions. However, the model with those features and without them did not change substantially. Nonetheless, as good practice the features exchange data and food inflation were dropped.



	Import Quantity	Export Quantity	Import Value	Exchange Data	Food Inflation	Crop Losses	Agricultural Land	Permanent Crops Area	Export Value
Import Quantity	1.00	0.26	0.76	-0.00	0.01	0.60	0.24	0.36	0.29
Export Quantity	0.26	1.00	0.33	-0.00	-0.00	0.42	0.26	0.13	0.74
Import Value	0.76	0.33	1.00	-0.00	0.01	0.56	0.33	0.39	0.51
Exchange Data	-0.00	-0.00	-0.00	1.00	-0.00	-0.00	0.01	0.03	-0.00
Food Inflation	0.01	-0.00	0.01	-0.00	1.00	-0.00	-0.00	0.00	-0.00
Crop Losses	0.60	0.42	0.56	-0.00	-0.00	1.00	0.41	0.50	0.33
Agricultural Land	0.24	0.26	0.33	0.01	-0.00	0.41	1.00	0.53	0.35
Permanent Crops Area	0.36	0.13	0.39	0.03	0.00	0.50	0.53	1.00	0.17
Export Value	0.29	0.74	0.51	-0.00	-0.00	0.33	0.35	0.17	1.00

Figure 1. Correlation matrix heat map