Predicting Global Food Security Indices

CS 6220 – Data Mining Techniques (Spring 2023)
Team Magnolia: Adrian Theodor, Binita Shakya, Sreelaya Devaguptam, Vanessa Chatman

Abstract

Global food security is a critical issue that affects the well-being of millions of people around the world. In this project, our team, Magnolia, worked to isolate the key features that determine and predict global food security indices. To accomplish this, we utilized the Global Food Security Index 2022 dataset, which includes food security index scores for 113 nations as well as determining factors grouped into the following categories: Affordability, Availability, Quality and Safety, and Sustainability and Adaptation. We employed four machine language models to isolate the set of features with the greatest predictive power: regression, decision trees, neural networks, and support vector machines. This report includes a summary of our data processing and analysis methodologies, a description of our principal findings, visualizations of our evaluations metrics, and a discussion of any relevant impacts our findings may have on global policy. Our results provide valuable insights into the factors that contribute to global food security and can inform policymakers on how to improve food security for the world's population.

Introduction

Food security is a critical global issue affecting millions of people worldwide. According to the United Nations, 811 million people worldwide were undernourished in 2020. The COVID-19 epidemic has aggravated the food insecurity situation, causing interruptions in food supply chains, job losses, and economic insecurity among vulnerable groups. Global food security is a complicated subject that necessitates a thorough understanding of the factors that contribute to it.

This project aims to discover the key factors that influence and predict global food security indices. Our research focused on the Global Food Security Index 2022 dataset, which provides food security index ratings for 113 countries as well as determining criteria classified as Affordability, Availability, Quality and Safety, and Sustainability and Adaptation. We attempted to isolate the collection of features with the greatest predictive value by using machine language models such as regression, decision trees, neural networks, and support vector machines.

Problem Description

The problem we aimed to address is global food insecurity, a complex and multifaceted challenge that affects millions of people worldwide. To identify the key features that contribute to global food security, we analyzed the Global Food Security Index 2022 dataset using machine language models. By providing policymakers with valuable insights, our analysis can inform strategies to improve food security for vulnerable populations.

Dataset

The data which we are using in order to train our models is the Global Food Security Index (GFSI). The dataset is developed by Economist Impact and sponsored by Corteva Agriscience. The GFSI considers and presents 91 determining factors for the overall index score of each country. The index assesses food security in 113 countries which span a diverse mix of income levels and climates.

The driving forces which contribute to a country's overall score have been divided into 4 pillars which are meant to better organize the factors which the dataset examines. These pillars are affordability, availability, quality/safety, and sustainability/adaptation. The pillar of affordability is meant to determine the degree to which people in a country can acquire nutritionally dense and low-cost food. This pillar includes aspects such as the quality of each country's food safety net programs and the percentage of citizens below the global poverty line.. The next pillar, availability, checks how diverse food source options are in each country as well as how easy it is for citizens to access food. This includes how a country empowers women farmers and governmental commitment to innovative technologies. The quality and safety pillar of the dataset determines the effect of the standards for food quality in each country. It also evaluates whether the country empowers its citizens to foster an environment for safe food based on aspects like access to clean drinking water and food safety measures. Lastly, the sustainability and adaptation pillar is used to find out if a country is properly preparing to tackle the climate crisis and how it can devastate food security if not well addressed. It includes factors such as temperature rise and agricultural water risk. Overall by combining the information from all four pillars the Global Food Security Index establishes itself as a useful means by which we can make more responsible governmental choices and ensure greater global food security.

Models

To identify the most influential features, we utilized a comprehensive approach by utilizing a variety of machine learning models, including regression, decision trees, neural networks, and support vector machines. These models were chosen due to their ability to handle complex data and identify significant patterns. By analyzing the output of each model, we were able to identify the set of features that had the most significant predictive power.

Linear Regression

Linear regression is a statistical method used to establish the relationship between a dependent variable and one or more independent variables. The goal of linear regression is to find the best fit line (i.e., a linear equation) that describes the relationship between the independent variable(s) and the dependent variable. Linear regression analysis is commonly used in many fields, including finance, marketing, engineering, and social sciences, to model and predict the relationship between variables.

Data Preprocessing:

In our Linear Regression analysis, the target variable or dependent variable of the dataset, which is already binned or discretized into categories. The dataset is then split into training and testing sets using the *train_test_split()* function from the scikit-learn library in Python.

Model:

We fit a linear regression model using all features of the training data and predict on the testing data.

Model Evaluation:

We have used different evaluation metrics such as Mean Squared Error (MSE) and R-squared to evaluate the performance of the model. The mean squared error (MSE) is a measure of how well the linear regression model fits the testing data. It is the average of the squared differences between the predicted values and the actual values.

In this case, the MSE value of **1.8761904690552602** means that on average, the predicted values of the model are about 1.88 units away from the actual values. The MSE can be used to compare the performance of different models, with lower values indicating a better fit to the data.

The mean absolute error (MAE) measures the average magnitude of the errors in a set of predictions without considering their direction (positive or negative). It is a commonly used metric in regression analysis to evaluate the performance of a model. The lower the MAE, the better the model's performance. This model has an Mean Absolute Error of **1.0794411993679158**.

The R-squared value is a measure of how well the regression model fits the data. It indicates the proportion of variance in the dependent variable that can be explained by the independent variables in the model. The R-squared value ranges from 0 to 1, where 0 indicates that the model does not explain any of the variance in the data, and 1 indicates that the model explains all of the variance. The R-squared value is also computed between the actual y values for the test set - y_test and the predicted y values - y_pred using the *r2_score()* function from the *sklearn.metrics* module.

The R-squared value of **0.3895648234652742** means that approximately **39%** of the variance in the target variable (y) can be explained by the independent variables (X) included in the linear regression model. This indicates that the model explains some, but not all, of the variation in the data.

Feature Importance:

The features are individually fit to the linear model to measure the feature importance. Based on the output, some of the features have a high coefficient and high variance score, indicating a strong positive correlation with the target variable. Specifically, the following countries have relatively high coefficients and variance scores: Algeria, Australia, Botswana, and Bahrain.

On the other hand, some countries have low coefficients and variance scores, indicating a weak correlation with the target variable. For example, Burkina Faso, Burundi, Cambodia, and Canada have low coefficients and variance scores close to zero or negative, suggesting that they are not good predictors of the target variable.

Some countries, such as Angola, Austria, Belarus, and Brazil, have low variance scores despite having non-zero coefficients. This suggests that these features may not be reliable predictors of the target variable, as they do not explain much of the variance in the target variable

Decision trees

Decision trees are a form of data mining technique where data is separated in a recursive way. The data points are split in terms of how homogenous they are with the target variable. It is for that reason why decision tree models are so effective at finding the most important features of a dataset. Once the tree is fit then the splits in the tree at each node can be examined to find the key features in the data. The farther a node is from the leaf layer, the more important the feature is in determining index score.

Processing Steps:

There is a Jupyter notebook in which a decision tree regression model was trained on our GFSI dataset. To begin the necessary Python libraries for this portion of our project were loaded into the notebook. Then, Excel files for ten years of the global food security index were imported into the notebook. Before loading the files into the notebook the sheets were cleared of features which were not relevant to our experiment. The stripped features include weight, mean, and the feature labels. The data was then checked for null values and reorganized to better work with SKLearn's DecisionTreeRegressor then transposed. The categorical data such as AVAILABILITY and AFFORDABILITY were removed as well as a few other features which we have found could be overly explanatory. Once the target values and features were defined then we used *sklearn train_test_split* to split the data so that the test data was 30 percent of the total dataset, and the training set was the other 70 percent.

Model:

A DecisionTreeRegressor was trained on the training set with no maximum depth specified. This tree was then tested against the test data and the R-squared scores were reported. Then we created a cell in the notebook which would iterate through a range of maximum depths to train, test, and report the R-Squared scores of DecisionTreeRegressor models for as many models as is equal to the max depth of the unpruned first tree model we trained. We did the same but for a different set of DecisionTreeRegressor models using mean squared error values rather than R-squared scores. The below plot shows that after six layers of depth, the accuracy score of each model plataus. This suggested that in order to prevent overfitting to the data we should train our model with a max depth equal to 6. If we increased the max depth there would be little to no increase in test accuracy. Finally, we saved the plot of the optimal DecisionTreeRegressor model.

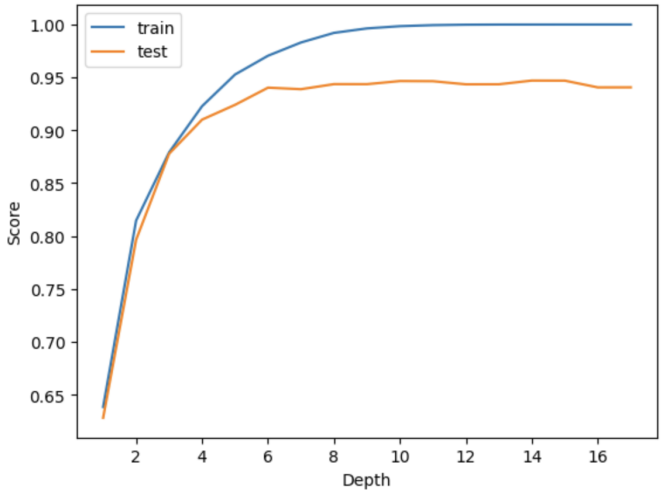


Fig. Train and Test Score against the Depth of the tree

Feature Importance:

Once we trained the DecisionTreeRegressor model we then sorted the feature importances determined by the SciKit library. The module finds the feature importances of the model by calculating the Gini score or the normalized reduction of the criterion for each feature. The mean squared error is the criteria we used for this model.

```
10 Most Important Features:
1. 1.2) Proportion of population under global poverty line (0.66725)
2. 2.8.2) Political stability risk (0.10925)
3. 3.5.3) Access to drinking water (0.08210)
4. 1.5.1) Presence of food safety net programmes (0.02708)
5. 4.5.1) Climate finance flows (0.02384)
6. 3.3.3) Dietary availability of zinc (0.02320)
7. 3.5.4) Ability to store food safely (0.01168)
8. 1.1) Change in average food costs (0.01107)
9. 1.4.2) Trade freedom (0.00654)
10. 4.5.6) Sustainable agriculture (0.00580)
```

Fig. Printing the ten most important features

Model Evaluation:

Our results from the DecisionTreeRegressor section of our project consistently indicate that the portion of a country's population who live beneath the global poverty line is the most important feature in determining GFSI score for each country, with a feature importance value of **0.66725**. Then we found that the feature importance for Political Stability Risk is the second highest at **0.10925**. Access to drinking water was determined to be the third most significant feature with an importance of **0.08210**. Other important features as determined by the DecisionTreeRegressor model include food safety net program availability, climate finance flows and ability to store food safely. The train MSE of the model we chose which is highly accurate but does not overfit is equal to **4.63** and a test score of **9.11** which suggests that this model where we selected a max depth of **6** is doing a fair job of generalizing and should work well with unseen data.

```
Tree Depth: <bound method BaseDecisionTree.get_depth of DecisionTreeRegressor(max_depth=6, random_state=42)>
Number of leaves: 60
Feature Importance: [1.10685486e-02 6.67254967e-01 0.00000000e+00 0.00000000e+00
 6.53817413e-03 2.70767938e-02 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.35086576e-03 8.87740957e-04 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.11072930e-03 1.22369455e-05
 7.19770339e-05 0.00000000e+00 2.83624967e-03 0.00000000e+00
 1.32131301e-03 2.64345127e-03 1.06380854e-04 0.00000000e+00
 2.18435658e-04 1.37256620e-03 2.89751211e-03 0.00000000e+00
 0.00000000e+00 1.09251180e-01 0.00000000e+00 1.32311114e-03
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.45806512e-03
 0.00000000e+00 0.00000000e+00 2.32017017e-02 6.79491296e-04
 2.00562370e-03 0.00000000e+00 8.21044661e-02 1.16828390e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 3.90090880e-04
 2.37700407e-04 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.82730084e-03 0.00000000e+00 0.00000000e+00 8.51733527e-04
 2.38364432e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 9.90366092e-04 5.79987010e-03 3.82075247e-03 3.77132186e-03]
Train score: 0.9704892971260461
Test score: 0.9403069390737027
Train MSE: 4.627128257744329
Test MSE: 9.105664199250842
```

Fig. Evaluation Summary

Neural network

The Global Security Index is a complex and multi-dimensional concept, and traditional statistical models may not be able to capture the complex relationships between the features and the target variable.

A neural network is a type of machine learning algorithm that is capable of learning complex patterns in the data by using multiple layers of interconnected neurons. Neural networks are well-suited for problems with high-dimensional data and non-linear relationships between the features and the target variable, making them a good fit for predicting the Global Security Index. The use of a neural network in this problem is justified by the complexity of the target variable and the need to capture nonlinear relationships between the features and the target variable.

Data Preprocessing:

The data is read from excel files using pandas library's *read_excel* function. The code renames the columns and sets it as the index of the dataframe. The code then drops the duplicate columns and rows in the dataframe.

The *preprocessingSteps* function is called to drop null values from X and y, transpose X, and convert y to a numpy array. The *splitingData* function splits the data into training, validation, and test sets using the *train_test_split* function from *sklearn* library. 80% of the data is used for training and validation, and 20% is used for testing.

We noticed that the data was not Gaussian distributed after plotting the features. As a result, we chose to normalize it instead of standardizing. To accomplish this, we utilized the NormalizingData function, which utilizes the StandardScaler from the *sklearn* library to normalize the data.

Model Architecture:

The neural network used in this code is a sequential model with **four layers of dense (or fully connected)** neurons. The first layer has **128** neurons, the second layer has 64 neurons, the third layer has 32 neurons, and the fourth (and last) layer has 1 neuron.

The activation function used for the hidden layers is **Rectified Linear Unit (ReLU)**, which is known to work well in deep learning models due to its ability to mitigate the vanishing gradient problem. The output layer doesn't have an activation function because it's a **regression problem** and we want to output a continuous value.

The optimizer used for this model is the **Adam optimizer** with a learning rate of **0.01**. The Adam optimizer is a popular choice for training deep neural networks because it combines the advantages of both stochastic gradient descent (SGD) and root mean square propagation (RMSprop).
The loss function used is **mean squared error (MSE)**, which is a common choice for regression problems. The model is also evaluated on **mean absolute error (MAE)** metric during training and testing.

```
Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_44 (Dense)            (None, 128)               11776

 dropout_33 (Dropout)        (None, 128)               0

 dense_45 (Dense)            (None, 64)                8256

 dropout_34 (Dropout)        (None, 64)                0

 dense_46 (Dense)            (None, 32)                2080

 dropout_35 (Dropout)        (None, 32)                0

 dense_47 (Dense)            (None, 1)                 33

=================================================================
Total params: 22,145
Trainable params: 22,145
Non-trainable params: 0
_________________________________________________________________
```

Fig. Model Summary

During training, the model is fit on the training set with a **batch size of 16 and for 1000 epochs**. The model is then evaluated on the validation set after each epoch to monitor overfitting. The plot of training and validation loss is generated using the *plotLoss()* function.

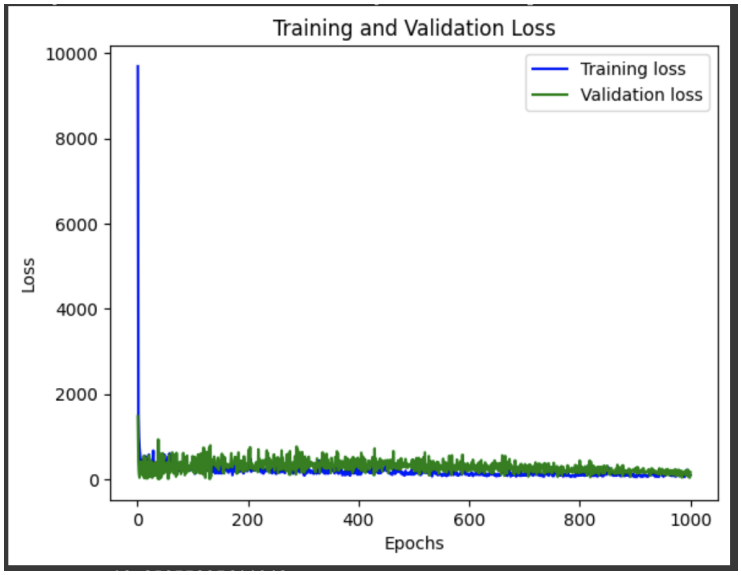


Fig. Training and Validation Loss vs Epochs (Year - 2020)

After training, the model is evaluated on the test set and the final MAE is printed. Additionally, the top 10 features with the highest importance are identified using the *top10Features()* function.

```
Top 10 features by importance in 2020:
1. 1.4) Agricultural trade
2. 4.5.4) Commitment to managing exposure
3. 3.1.2) Share of sugar consumption
4. 1.5.2) Funding for food safety net programmes
5. 2.2.2) Access to agricultural technology, education and resources
6. 2.7) Sufficiency of supply
7. 1.4.1) Agricultural import tariffs
8. 3.5.4) Ability to store food safely
9. 3.2.1) National dietary guidelines
10. 1.2) Proportion of population under global poverty line
```

Fig. Top 10 features by importance in 2020

Feature Importance:

The *top10Features* function calculates the feature importance of the trained neural network model and prints the top 10 features with the highest importance. The function uses the absolute sum of the weights in the first dense layer to calculate the feature importance.

Model Evaluation:

The code reads data from multiple excel files and trains a neural network model on each of them. For each file, generates a neural network model, prints the MAE, the top 10 features influencing the GSI, and adds the top 10 features to a list.

The code then flattens the list of top 10 features, counts the occurrences of each feature, and sorts them based on the count. The code then generates a bar plot of the top 5 most common features among the top 10 features influencing the GSI.

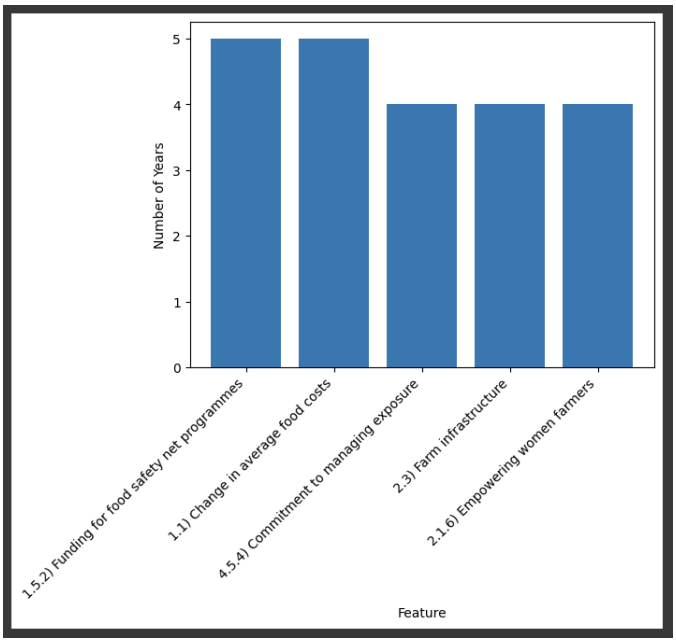


Fig. Among the top 10 features influencing the Global Security Index each year, the features that consistently appeared across most years.

The mean MAE for all years was calculated to be *3.02*, which is a favorable score for the target value range. Moreover, in terms of the top 10 features that influenced the Global Food Security Index between 2012 and 2022, "Funding for Food Security Net Programs" was the most prominent, appearing in 5 of those years, along with "Change in Average Food Costs." "Empowering Women Farmers," "Commitment to Managing Exposure," and "Farm Infrastructure" were the second most significant features, each appearing in 4 of the years.

Support Vector Machines

Support Vector Machines (SVMs) are machine learning algorithms that utilize support vectors (points in a dataset that lie closest to the model's decision boundary). While SVMs are used for classification tasks, however, our dataset features continuous data and not discrete targets, and thus this section explains our implemented Support Vector Regression (SVR), which is a machine learning algorithm that utilizes support vectors for regression tasks. SVR works by finding a hyperplane, which acts as a multidimensional regression line to which the model attempts to fit as the data points with the lowest margin of error possible, minimizing the difference between the predicted and actual values. Where Support Vector Regression differs from linear regression is that linear regression always assumes the relationship between the input and ground truth variables is linear, while SVR allows for a nonlinear relationship. Thus SVR analysis allows for a more complex relationship found within the data. It is helpful to note as well that the SVR model transforms the input variables into the high-dimensional space via a kernel function which allows the data to be separable via hyperplane.

Data Preprocessing:

Processing the GFSI data for SVR analysis required relatively straightforward cleaning and transformation of the data spreadsheets. Firstly, the dataset was transposed so that each country and its corresponding indices composed the data instances and the indices' columns composed the features. Next it is necessary to strip select rows and columns from the now transposed dataset. The rows with values such as the weight, average, minimum, maximum, etc., of the input index values should be stripped. Please note that this can be (and was) completed manually within the spreadsheet prior to import and transposal. Next, and importantly, all categorical features were stripped. These columns include major categories such as AFFORDABILITY and QUALITY AND SAFETY and even lesser, more explanatory categories such as 'farm infrastructure' and 'dietary diversity'. As aggregate categories of other dataset components, these columns were removed so as not to over-represent the same features. Lastly, the column FOOD SECURITY ENVIRONMENT was removed from each year of the dataset to be included separately as the dataset of target values.

Model:

Two SVR models were trained in this study. In the first, the data is split into the training and test sets, and sizes 80% and 20% respectively, and a random state of 42, using scikit-learn. The data then trained an SVR model using a linear kernel and scikit-learn's SVR function. The model was then fit to the training data and used to predict the target variable for the test data (y_test). Finally, the mean absolute error (MAE), mean squared error (MSE) and R-Squared score were calculated to evaluate the model's accuracy.

A note on the use of the linear kernel in the SVR model:

As mentioned, the kernel function of the SVR model allows the inputs to be transformed into the high-dimensional plane where it can be separated by the hyperplane. The kernel function is one of three

parameters of the SVR model that can be tuned, along with C, the regularization parameter which adjusts the trade of between the amount of training errors allowed and the amount of over-or underfitting (smaller C → more training errors but better generalization, larger C → fewer training errors but higher tendency to overfit), and epsilon, which sets the margin of tolerance for errors (smaller epsilon → smaller margin of errors and therefore more complex model, larger epsilon → greater margin of errors and therefore smoother, less complex model). The kernel parameters are linear, which represents and assumes a linear relationship between input target variables, polynomial, which assumes a polynomial relationship between the same variables, and radial basis function or RBF, which models complex relationships between input and target variables which are neither linear nor polynomial. Due to the options in parameters, sklearn's model selection function GridSearchCV was implemented to find the optimal, C, epsilon, and kernel type values for each of the years' data. Overwhelming the values, C=0.1, epsilon = 0.1, and the linear kernel were reported to be the best parameters for the overall data set. As such, these parameters were used in model fitting and feature selection.

Lastly, Recursive Feature Elimination (RFE) within sklearn.feature_selection was used to train a model again to yield ranking of the input features from least to most important. Once these features were ranked, this data was sorted, filtered, and examined in terms of ubiquity to find which feature indices best determined the food security index target.

Model Evaluation:

The initial SVR model fit with a linear kernel, C=0.1 and epsilon = 0.1 (default), yielded the following mean absolute error (MAE), mean squared error (MSE) and R-squared scores for each of the years of study:

```
2012
Mean squared error: 0.043526492559582776
Mean absolute error: 0.1633907821662801
R2 score: 0.9996878334844892

2013
Mean squared error: 0.059041385362943975
Mean absolute error: 0.19988873508745109
R2 score: 0.9995801695342746

2014
Mean squared error: 0.104565697380785
Mean absolute error: 0.26669347603078786
R2 score: 0.9992231623388129

2015
Mean squared error: 0.09482991057331283
Mean absolute error: 0.24210571777679582
R2 score: 0.9991678595247773
2016
Mean squared error: 0.12207003525056619
Mean absolute error: 0.27681617136631453
R2 score: 0.999051711827409

2017
Mean squared error: 0.1413603198785274
Mean absolute error: 0.3097195470276268
R2 score: 0.9990543551458025
```

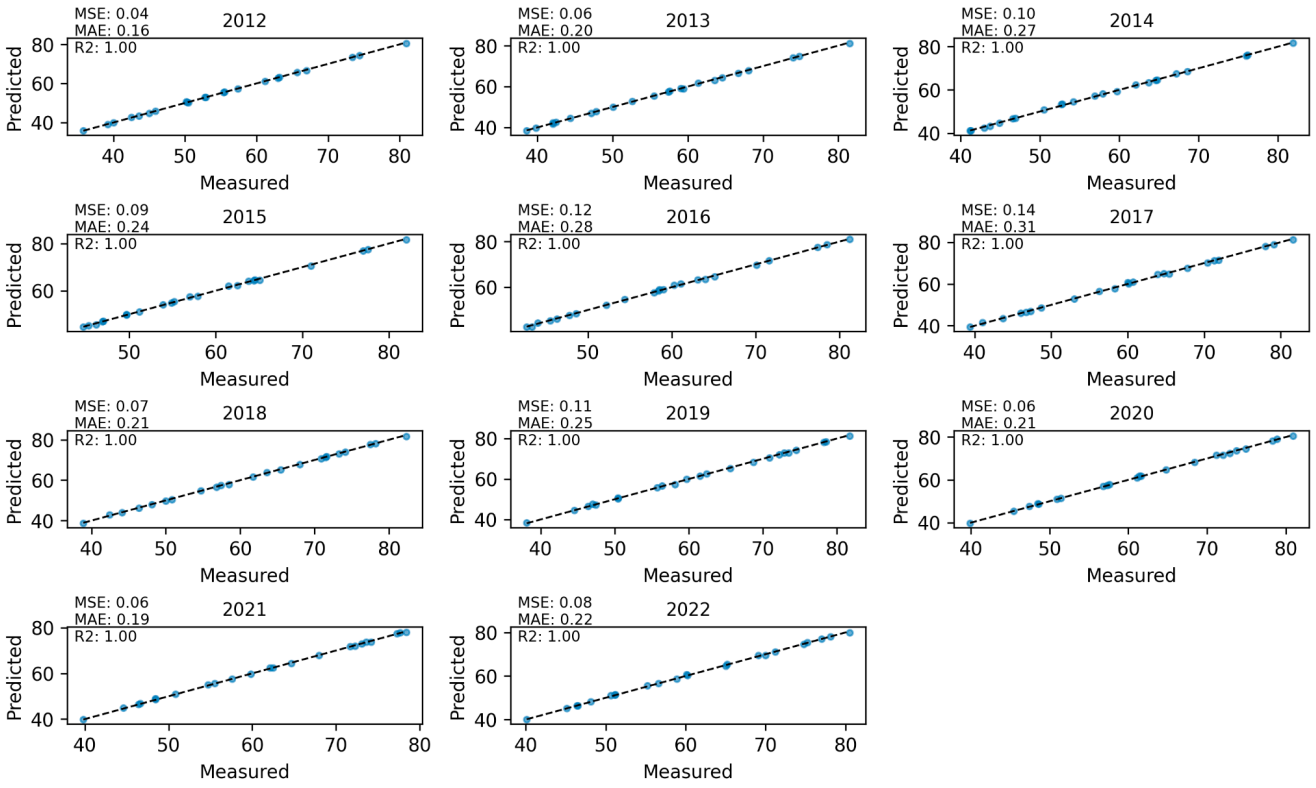
```
2018
Mean squared error: 0.07219712285653208
Mean absolute error: 0.21276949082157004
R2 score: 0.9995311338739671

2019
Mean squared error: 0.10950138571536774
Mean absolute error: 0.2543785934788199
R2 score: 0.9992774670430523
2020
Mean squared error: 0.062344956565797
Mean absolute error: 0.20797352167843225
R2 score: 0.9995530407149387

2021
Mean squared error: 0.05925101406941673
Mean absolute error: 0.19206929971903935
R2 score: 0.9995841811249165

2022
Mean squared error: 0.08235776596905656
Mean absolute error: 0.22130293992387814
R2 score: 0.9994159977335797
```

As the low mean errors and high R-squared suggest, a plot of the actual versus predicted values reveals the following data points fitted nearly exactly to the line of perfect accuracy:



Each point on these plots represents a single data point from the yearly test sets, where the x-coordinate is the actual value of the target variable and the y-coordinate is the predicted value of the target variable. The diagonal line on each plot represents the line where the actual values are equal to the predicted values. The black dashed line is drawn on this diagonal line to help visualize how well the predicted values match the actual values.

The SVR hyperparameter tuning for each year used Grid Search Cross Validation to find the best combination of hyperparameters for each year of data. The hyperparameters tuned were the kernel type (linear, RBF, or polynomial), the regularization parameter C (0.1, 1, or 10), and the epsilon parameter (0.01, 0.1, or 1). The best hyperparameters and the corresponding mean squared error (MSE) and R-squared (R2) score on the test data are printed for each year of data, providing insight on how to improve the accuracy of the model and make better predictions for food security indices:

```
Results for 2012_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.002888932219272052
R2 score: 0.9999792809424467

Results for 2013_Sheet1:
Best parameters: {'C': 1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.01383081926684187
R2 score: 0.9999016520486694

Results for 2014_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.0038866898596231643
R2 score: 0.9999711250071262

Results for 2015_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.005839799444287753
R2 score: 0.9999487552666095

Results for 2016_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.005685299600316967
R2 score: 0.999955834350686

Results for 2017_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.0045078153992862595
R2 score: 0.9999698444907336

Results for 2018_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.00540645530132503
R2 score: 0.9999648891305857

Results for 2019_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.007741758696128087
R2 score: 0.9999489168491691

Results for 2020_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.005609269899500508
R2 score: 0.999959786398097

Results for 2021_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.0025805007519108924
R2 score: 0.9999818902522317

Results for 2022_Sheet1:
Best parameters: {'C': 0.1, 'epsilon': 0.01, 'kernel': 'linear'}
Mean squared error: 0.003453212278911894
R2 score: 0.9999755131313534
```

Lastly, The RFE object is fit to the input variable matrix and target variable vector to select the best features based on their importance in predicting the target variable. The resulting feature selection for all yearly data was then sorted, filtered, and summed in terms of greatest presence across the years to reveal the following most key features:

Feature: 2012-2022 top ten most predicted feature set appearance count

{
 "1.4.2) Trade freedom": 9,
 "3.5.3) Access to drinking water": 7,
 "1.2) Proportion of population under global poverty line": 6,
 "4.2.1) Agricultural water risk \u2013 quantity": 6,
 "1.3) Inequality-adjusted income index": 6,
 "1.5.1) Presence of food safety net programmes": 5,
 "3.3.3) Dietary availability of zinc": 4,
 "2.7.1) Food supply adequacy": 4,

"4.5.4) Commitment to managing exposure": 4,
"2.6.1) Planning and logistics": 4
}

Results

Conclusions to be drawn from a comparative analysis of the mean absolute error given by the Linear Regression, Neural Network, and Support Vector Regression models show that the SVR yield the lowest MAE scores (whose average error for all years is < 1.00 compared to NN's 3.02 across all years and LR's 1.08 for 2020.

Conclusions to be drawn from a comparative analysis of the mean squared error given by the SVR model and Decision Tree also show a lower MSE yielded by the SVR versus the DT (< 1.00 and ~9.10 respectively).

Discussion

From Linear Regression, we realized that other techniques such as feature selection methods that consider the interactions between features or regularization methods must be used to address overfitting.

The results as produced during our DecisionTreeRegression model made logical sense to our team. The conclusion we arrived at was that according to this model the portion of a population below the global poverty line is the most important feature in this set. This was not shocking to us as the authors of the Global Food Security Index mentioned it was one key feature to examine more closely. They also mentioned that the rising cost of food was becoming more important in recent years which could explain why it was in the top 10 important features as found by this model.

Our neural network model identified funding for food safety net program availability as the most significant factor affecting food security. The model also highlights the importance of access to empowering women farmers and farm infrastructure in determining food security.

Our SVM model helped us recognize the importance of trade freedom, access to drinking water, and proportion of population under the global poverty line in predicting the Global Security Index.

Our findings have important implications for policymakers seeking to improve food security. Our models provide valuable insights into the factors that contribute to global food security and can help inform policymakers on how to improve food security for the world's population.

Overall, our analysis suggests that a combination of factors must be addressed to improve global food security. Addressing the percentage of individuals living below the global poverty threshold, fostering political stability, enhancing the availability of uncontaminated water, and implementing food safety measures are crucial initiatives that must be undertaken. Our machine learning models provide an innovative way to analyze large datasets and can help policymakers make informed decisions to address food security issues.

Future Work

Our work could be updated every year in which a new index is debuted which could increase the accuracy of our models. Or a dashboard could be made which allows users such as public servants to switch between the different models we have trained and use them in order to track food security trends or challenges. We have determined decisively that by far the most important feature is the portion of the population of a country under the global poverty line. It would be interesting to focus our attention on this portion of the global food security problem. In what exact ways does poverty change food access and how can we measure them? That work could lead to better understanding of the tether tying food insecurity and poverty. Another option for further work could be to investigate where the highest scoring countries are excelling and determine how to use them as a role model for other countries.

References

Crop Recommendation and Yield Prediction for Agriculture using Data Mining Techniques
Crop Suggestion using Data Mining Approaches
Proposed framework for implementing data mining techniques to enhance decisions in agricultural sector
Utilization of Data Mining Techniques in National Food Security during the COVID-19 Pandemic in Indonesia
Prediction of Crop Production in India using Data Mining Techniques
Comparative Study of Data Mining Techniques in Crop Yield Production
Environment Monitoring System for Agricultural Application using IoT and Predicting Crop Yield using Various Data Mining Techniques
Climate, Agriculture and Hunger: statistical prediction of undernourishment using non-linear regression and data-mining techniques
Tracking Food Insecurity from Tweets Using Data Mining Techniques

Annex-A

Adrian Theodor: Decision Tree Code and explanation, Part of Discussion, Part of Future Work, Description of the Data set.
Binita Shakya: Linear Regression
Sreelaya Devaguptam: Neural Networks, Discussion, Formatting the entire document
Vanessa Chatman: Support Vector Regression, Results, Contributions to Abstract and Introduction