# Applied Data Science Report

Dimitrios Pilitsis
*Department of Computer Science*
*University of Bristol*
Bristol, United Kingdom
ka18772@bristol.ac.uk

William Nafack
*Department of Computer Science*
*University of Bristol*
Bristol, United Kingdom
dc18128@bristol.ac.uk

Theodore Clarke
*Department of Computer Science*
*University of Bristol*
Bristol, United Kingdom
bf18890@bristol.ac.uk

Ishaq Gunawan
*Department of Electrical and Electronic Engineering*
*University of Bristol*
Bristol, United Kingdom
ou18497@bristol.ac.uk

Kenneth Tan
*Department of Electrical and Electronic Engineering*
*University of Bristol*
Bristol, United Kingdom
wb18936@bristol.ac.uk

***Abstract*—This report investigates food insecurity of over 35,000 households located across 33 different countries by analyzing a subset of the over 800 variables that are provided with each survey. The aim of the investigation was to create a model that is as accurate as possible in predicting the food insecurity level of a given survey, while also determining the main factors that affect food insecurity. We began by cleaning the dataset and then performing Exploratory Data Analysis to better understand the dataset. We proceeded with feature selection which showed that using the entire dataset results in the highest accuracy. We then investigated a wide range of machine learning models and then focused on support vector machines, feedforward neural networks and random forests. The aforementioned models obtained accuracies of 65.2%, 65.0% and 70.5%, respectively. Our results are helpful in understanding the power of asset based indicators such as the Poverty Probability Index in predicting food insecurity, and that advanced machine learning techniques are superior to basic logistic regression models for a problem like this.**

## I. Introduction

Twenty five thousand people die every single day due to hunger [1]. Some 850 million people are estimated to be undernourished in 2021 [1], lacking consistent access to adequate amounts of food to allow them to live an active, healthy lifestyle [2]. Smallholder farmers and their families face particular difficulty, representing around 2 billion people of the global population. An estimated 85% of farms (around 450 million) worldwide measure less than 2 hectares, and the average farm size is continuing to decrease. The majority of smallholder farmers and farm workers earn less than $2 per day. Most of them have to buy more food than they are able to produce [1].

There are many reasons why food insecurity is such a prevalent issue in our society, including but not limited to: food prices, lack of access to land and the quality of land. All these factors effect the income of a farmer and the yield of their farm, causing great difficulty in farmers providing for their families. Food security is such a dire problem that the United Nations has made it their 2nd goal - to end world hunger. In 2015, all United Nation members adopted the 2030 Agenda for Sustainable Development, a blueprint for peace and prosperity [3]. The Rural Household Multi-Indicator Survey (RHoMiS) is an excellent dataset that has aggregated multiple surveys into one large dataset that contains over thirty five thousand observations from thirty three countries [4]. The dataset contains over eight hundred variables, ranging from the Poverty Probability Index (PPI) likelihood, which estimates the probability that a respondent is above/below the poverty line, to the amount of land that is owned by a farmer.

Being able to predict and understand how these variables affect food security could lead to a better understanding of world hunger and how to eradicate it - be that in the form of government legislation or optimizing charity aid to poverty stricken regions. The latter is especially of importance, as for decades charities have been directing aid based on the output of overly simplistic early warning systems which produce questionable results, as they rely on very few variables that have historically been used, such as household income or land owned. As we will see in this investigation, food insecurity depends on a multitude of variables and the relationships are extremely non-linear, signifying that the methods currently used are potentially flawed and have lots of room for improvement. Understanding which important factors actually affect food insecurity and the relationships between them could lead to better allocation of resources and capital for farmers who need them.

Predicting food insecurity has existed as a task for many years, but has mainly focused on a small subset of methodologies, primarily logistic regression [5]. It has brought about limited progress in accurately predicting food insecurity [6][7], as they focus on model interpretability. Hence, we decided to implement advanced Machine Learning models to determine if we could obtain better accuracy than other publications have. This leads us to the main aim of this project, at the request of the end user - to create a model that is as accurate as possible in predicting the food insecurity level of a given survey. Our secondary aim is to determine which features are of utmost significance in predicting food insecurity and whether they

coincide with the conventional features that are considered significant for predicting food insecurity.

To reach such conclusions, we will begin by understanding how we prepared the dataset in order for it to be used by a model. We will then proceed by viewing our initial thoughts of the dataset after performing Exploratory Data Analysis. This will then lead us to what methods of feature selection were used to be inputted into our models. We will then analyze the range of models we investigated and their accompanying results. We will finally evaluate our findings and draw conclusions based on our analysis.

## II. Data Preparation

Real world data is known to often be messy and noisy. Due to the scope of the project, a subset of the 800 variables had to be analyzed due to time constraints, thus utilizing a condensed dataset provided by the data owners. They provided us with a csv file, having already handled data privacy concerns to ensure no individual survey is uniquely identifiable. Initially, the RHoMIS dataset was particularly challenging to deal with due to the large amount of missing data. This is due to the dataset being the result of aggregating multiple surveys. Moreover, after communicating with the problem owner, we realized international survey efforts face a range of challenges which affect data quality, these include: translation and harmonisation, non-standard units and recall error. The aforementioned points warranted significant imputation in order to perform any data analysis. Figure 1 shows the significant amount of missing data for a sample of features from the dataset (all code can be found on GitHub repository [8]).

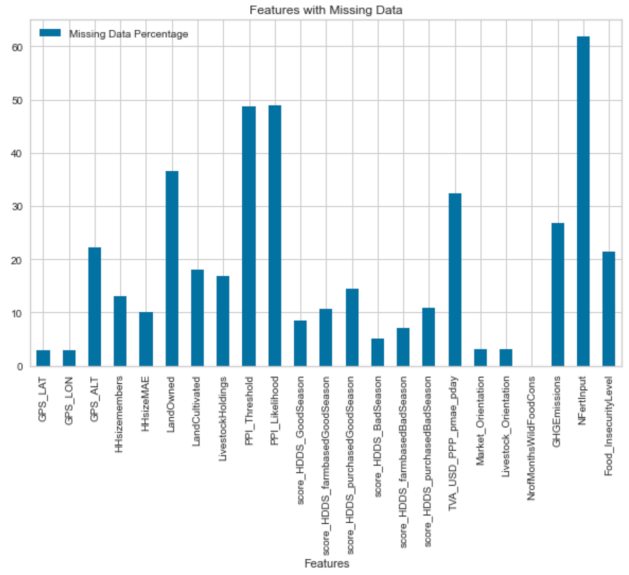


Fig. 1. Bar chart displaying the amount of missing data for a subset of features of the dataset.

### A. Simplifying the food insecurity labels

The dataset provided us with two food insecurity indicators, namely the Household Food Insecurity Access Scale (HFIAS) and the Food Insecurity Experience Scale (FIES). Scores are based on the answer of eight questions, of which people answer them according to the severity of the food insecurity they experience. The scores range from severely food insecure to food secure. We learnt that FIES was created to replace HFIAS in 2019, which allowed us to combine the scores together. This meant we could use all thirty five thousand observations to train a single model instead of splitting into two.

After deep research, a complication of the FIES score was discovered. The relative severity of each question causes huge variation in the answers provided. In different countries, or even sub-populations, the relative severity associated with each score may vary. One reason is due to the nuances in the translation, meaning that the same question can be interpreted differently depending on the interviewee. In addition, food insecurity conditions are experienced and dealt with differently based on culture and livelihood systems. In order to analyze the results, the data owners utilized a Rasch model [9]. The raw score of FIES is between zero and eight, yet it only provides an ordinal measure of food insecurity. The difference in severity between adjacent raw scores is not constant e.g. a raw score of four is more food insecure than someone with a raw score of two but is not twice as food insecure. To deal with this issue, we decided to perform discrete assignment on the labels, creating a conventional 4-label classification task, using the following map for food insecurity (FI) [10]:

- Severely Food Insecure: $\{7, 8\} \rightarrow 4$
- Moderately Food Insecure: $\{4, 5, 6\} \rightarrow 3$
- Mildly Food Insecure: $\{2, 3\} \rightarrow 2$
- Food secure: values $\{0, 1\} \rightarrow 1$

### B. Data Cleaning

Before any imputation could take place, the data had to be cleaned and prepared. A multitude of methods were used to clean the dataset. Firstly, we dropped the following features as we did not need them for analysis as they were duplicate features: ID features, Regions, currency conversion and year. Secondly, we had to translate some of the entries of months into English as they failed to be translated when the survey was made. Additionally, some categorical features such as education level had typos or duplicates which meant that they had to be remapped. Moreover, after consulting with the project owner, we realized that many features had negative values even though they were not meant to be negative such as land owned. This resulted in us replacing these negative values with "Not a Number" so that they could later be imputed. As with any real dataset, outliers are present. We decided to define an outlier as "a point which falls more than 1.5 times the interquartile range above the third quartile or below the first quartile" [11]. Most outliers were set to "Not a Number" so that they could be imputed, while a handful of values were too extreme (e.g. value is 10 billion when the average value for the feature is 0.01) so we removed the survey entry.

### C. Imputation

After cleaning the dataset we were able to proceed with imputation. Imputation was especially important given the amount of missing data. Our initial thoughts were to implement as many imputation methods to see which ones were suitable. We investigated: case deletion, pairwise deletion, average/median/mode imputation, regression imputation, impute from nearest neighbour, Expectation Maximization, a variant of Multivariate Imputation by Chained Equations (MICE) and Decision Trees. It was determined that the simpler methods did not suffice, for example, case deletion left us with only 0.24% of the original dataset. We decided to use a technique where we model each feature with missing values as a function of other features, and then use that estimate for imputation. This is performed iteratively, akin to round-robin, known as Multivariate Imputation by Chained Equations (MICE) [12], although we used Sklearn's [13] Iterative Imputer. At each step a feature column is chosen as the output vector and all the other features are treated as input vectors. A model is then fit on the aforementioned dataset which is then used to predict the missing values. This is completed for each feature iteratively. We opted to use the random forest regressor to impute the data as it generally helped retain the distribution of the different features. An example can be seen in Figure 2, random forest does a much better job at keeping the mean and bounds of the original distribution of PPI compared to regression imputation.

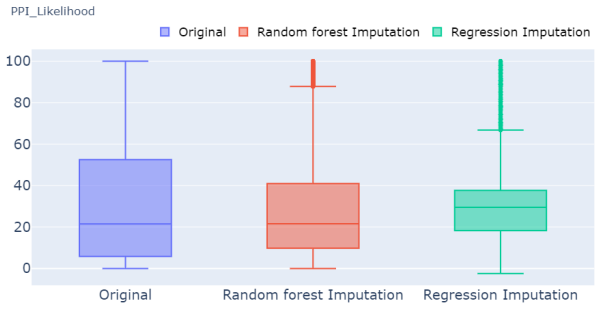


Fig. 2. Box plots displaying how Random forest and regression imputation changed the original distribution of PPI likelihood through imputation.

### D. Encoding

After imputation, we then had to encode categorical features. One hot encoding was used for all features except for months. Months was particularly interesting to encode due to its inherent cyclical nature. Months such as December and January need to be close to one another when modelling the data. We decided to use cyclic encoding as it provided an innovative way of handling seasons. We created two features for the month, one takes the sine of the month index, multiply it by two pi and the another where you take the cosine of the same. This way, one can uniquely identify a month when solving the equations while also encoding the cyclical nature of months. The results of cyclic encoding proved to be superior compared to one hot encoding and was selected to be used as the preferred method for encoding months.

### E. Data normalization

After the aforementioned techniques were applied, the last method to apply was data normalization. Scaling data is extremely significant as it improves the performance of optimization based techniques while aiding methods where the coefficients of the model are directly related to the magnitude of the feature. A minimum maximum scaler was selected as it makes no assumptions on the distribution of features, unlike standardization [14], while also being quicker to compute.

## III. Data Exploration

After imputating the dataset, exploratory data analysis needed to be performed in order to determine how much of an effect imputation had on the original dataset.

### A. Correlation of features

After removing all outliers, we decided to determine which features correlate with Food Insecurity Level. No single feature has a stronger correlation than 0.3 with food insecurity, as seen in Figure 3. This was extremely surprising as our initial thoughts were that the number of months food insecure, total income or Poverty Probability Index (PPI) likelihood would have had a very strong correlation with the Food Insecurity Level label. This finding only highlights how complex the task at hand is, as if there are relationships to be learned from the data, then they are likely complex and non-linear.

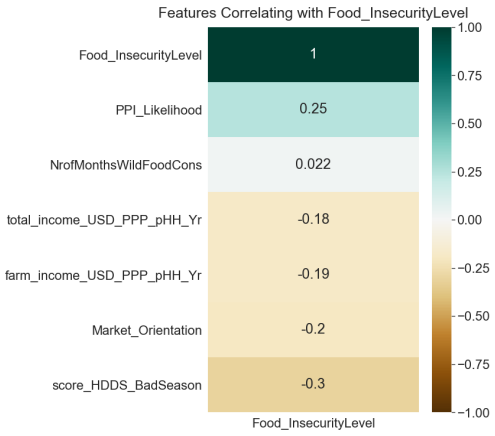


Fig. 3. Correlation heatmap for the main features of the dataset with respect to the label Food Insecurity Level.

We then decided we potentially wanted to reduce the number of features we use before moving onto feature selection by using correlation heatmaps. We set a threshold of $\pm 0.7$ to determine whether or not to discard a feature that is heavily correlated with another feature. This is done as two features that are heavily correlated often do not increase the predictive power of a model and only adds to the dimensionality of the data, allowing overfitting to possibly occur. We then decided to determine the relationships of specific variables to see if

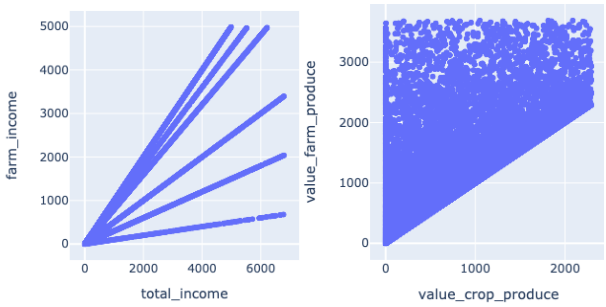


Fig. 4. Plot displaying total income vs farm income (left) and a plot displaying total income vs farm income (right).

they were linear or non-linear. As seen in Figure 4 , there is a clear linear relationship between farm income and total income, while for others features such as value crop produce and value farm produce they are non-linear.

We then want to see the distribution of the observations with respect to the label, which can be seen in Figure 5. We see that the distribution of the labels are not equal; the Mildly Food Insecure label has half of the number of food secure label. This is an imbalanced data set and it is possible that there aren't enough Mildy FI datapoints for the model to properly learn how to classify them.

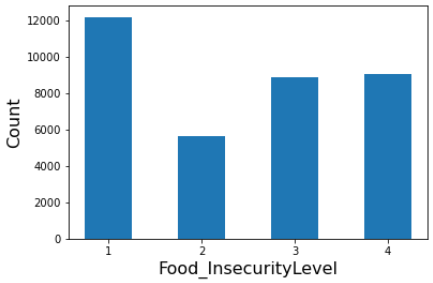


Fig. 5. Bar chart displaying the distribution of labels amongst observations.

### B. Dimensionality reduction

Due to the dataset we were working with having 96 variables, we decided to try some dimensionality reduction techniques to see if we could understand the data better. We began with Principle Component Analysis[14]. The principle component only contained 16.1% of the variance of the data, followed by the second principle component with 7.7%. Together they are under 25% of the entire variance of the dataset, suggesting that PCA was not able to separate the features well. We then decided to create a visualization after applying dimensionality reduction techniques, so naturally a Uniform Manifold Approximation and Projection (UMAP) was chosen [15]. UMAP is a dimensionality reduction technique used for visualizations, similar to t-SNE, but is primarily used for general non linear reduction. Unfortunately, the labels were dispersed around mini clusters with no specific pattern.

## IV. Feature Selection

Before finally moving onto modelling, we first need to implement methods to systematically decide what features are desirable to be used for a model. There are a multitude of reasons why one uses feature selection. Firstly, by using too many features, the models can overfit and not generalize well. Also, any features that aren't useful that weren't removed during data preprocessing will be discarded using feature selection.

We decided to perform a selection of $k$ features using multiple methods: chi-squared test, mutual information and recursive feature elimination. The former two methods are used as a univariate estimator on each feature to obtain scores. Then, to obtain the top $k$ features, we take the features with the $k$ highest scores. Recursive feature elimination uses an estimator to assign weights to all the features and then recursively reduces the number of features we take into consideration. We initially train the estimator on all the features and determine each feature's importance. We then prune the least important features from the set of all features and recursively do so until we have a set of $k$ features. Let $k \in \{20, 30, 40, 50, 60, 70, 96\}$, where $k$ is the number of features retained after feature selection. For each model type and for each label of the target variable, we obtained: an accuracy score, precision, recall, f1 score, and support. We also decided to use a logistic regression model as a baseline feature selection estimator before deciding to use any more advanced techniques. It is worth noting however, that different algorithms do make use of different combinations of features different and that model specific experimentation is also important.

Accuracy is simply the ratio of the correct predicts to the total number of observations. Accuracy works well in situations where there are an equal number of observations per label, otherwise the accuracy score can be heavily skewed and not provide an understanding of the data. As we do have skewed labels, we decided to also analyze the F1 scores of the models. The F1 Score is simply the harmonic mean of precision and recall, where precision $= \frac{\text{True Positives}}{\text{True Positives + False Positives}}$ and recall $= \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$ [14]. A high precision value, but a low recall value signifies that the model is accurate but misses a significant number of observations that are considered difficult to classify. The main use of the F1 score is in informing how many observations are correctly classified, while also explaining how often the model misclassified an observation. The aim is to maximize F1 scores. We analyze how the number of features retained affect the F1 score, seen in Figure 6.

Interestingly, we see a strong and consistent positive relationship between $k$ and the F1 score, while we see that the different methods overall perform similarly. The f1-score never seems to converge to a specific value, even as $k$ tends to 96 features, indicating that for predictive accuracy, all features would be necessary. This shows that in order for a model to obtain the highest accuracy, we need to utilize all the features, hence, we will use all features when creating the models unless we are trying to reduce computation needed. It also implies that individual features on their own are not that useful, but

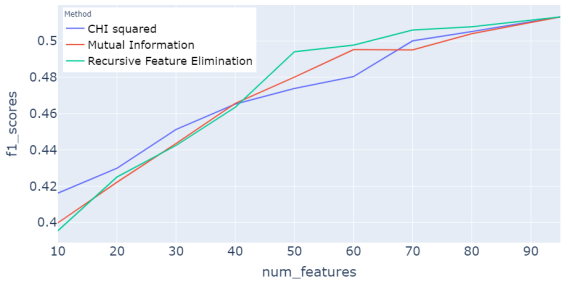


Fig. 6. Plot displaying how the number of features kept from feature selection affects the F1 score when used with a logistic regressor for the three techniques.

combining them together provides a lot more possibilities for producing practical models.

## V. Modelling, Results and Analysis

Exploring different classification algorithms when modelling is an incredibly time-consuming process. There are many algorithms that are suitable for different contexts, with different strengths and weaknesses. Therefore, it is important to choose algorithms based on their suitability in terms of the dataset and the aims of an investigation.

The EDA already revealed that the dataset is considerably noisy. Also, the lack of significant correlations of individual features with Food Insecurity Level suggests that if there are underlying relationships between the features and the Food Insecurity Level, it will take appropriately complex models to effectively learn them because they are likely to be non-linear. The noise displayed within the dataset and the potential underlying complexity suggests that this challenge can be viewed in terms of the fundamental supervised learning problem: "The Bias-Variance Trade-off". Training a classifier to accurately predict Food Insecurity will require a model that doesn't have too much bias i.e. being simple enough to model the true relationship between the features and the target, or too much variance, where the model is overtly complex and learns the noise in the data. This is otherwise known as underfitting and overfitting, respectively [16].

Machine learning is a field full of trade-offs and many modelling decisions made were the result of them. The primary aim of the investigation is to predict food insecurity as accurately as possible, but it is worth considering the fact that methods that lead to the most accuracy, can be less interpretable. Furthermore, time and compute power are inherently limited and this can get in the way of finding the most accurate model. Although accuracy is the priority, it is worth considering each of these factors. Finally, it is important to take into consideration the inherent Bayes Error i.e. the lowest possible error that can be obtained by the classifier which is due to the features not containing all the information about the output variable and the intrinsic stochasticity of the system [16].

Given the aforementioned points, we begin with our hypotheses. we hypothesize that simple models such as: logistic regression, support vector machine with a linear kernel or a stochastic gradient descent classifier will have too much bias to accurately capture the relationships. Instead, we conjecture that methods such as: support vector machines with a higher dimensional kernel, neural networks or sensitive tree-based methods are more likely to capture the highly complex underlying relationships, while also employing different methods to prevent overfitting, either utilizing regularisation or, using diversity in an ensemble method to reduce variance.

The models developed will be assessed using the following metrics:

- Accuracy Score: The proportion of predicted labels that match their true label.
- F1 Score: The harmonic mean of precision and recall.
- Shapley Values: A measure of contributions each feature has in a machine learning model, based on a game theoretic approach to explain the output of any machine learning mode. [17].

The accuracy score is a simple metric that can be useful when the classes are equally distributed to notify one for a prediction, how likely is the correct classification. F1 score is slightly more complex as it can be used to judge, in a class of predictions from a model, the proportion of predictions in a class that actually belong to that class (precision) as well as for a given class, the proportion of actual times an entry from said class is classified correctly (recall) in tandem. This has huge implications in this context, as given the principle aim of this investigation, a model could be employed to make decisions about where to direct aid based on indicators, or it can be used to underpin an early-warning system to try detecting food insecurity before a period of increased food insecurity e.g. drought, economic recession. This has real consequences. Therefore, individual F1 scores for each class are extremely important, particularly the more severe classes (moderately and severely FI - 3 and 4, respectively). Misclassifying a future data point, which in reality is a small-scale farmer, for example as food secure even though the farmer is severely food insecure, can have grave consequences on whether they receive the appropriate aid that they need.

Shapley values and plots are crucial because is likely that incredibly complex, black-box models will be needed to solve this problem, but actually it is a problem where interpretability is key. Learning about what contributes to food insecurity is just as important as predicting it and Shapley plots allow for both goals to be met.

### A. *Initial findings*

It is important to not only make decisions based on our assumptions made about the data, but to also experiment and determine what methods work and whether our assumptions do indeed hold. As part of the early exploration, we begin by deploying a large range of classifiers using default hyperparameters [13] to determine which methods clearly showed potential to be explored further by performing hyperparameter tuning and possible feature selection. Those that clearly performed poorly were discarded. After scaling the data using the minimum

maximum scaler seen in the previous section, and splitting the dataset 75:25 into training and testing sets, respectively, we obtained the accuracy results which can be seen in Figure 7.

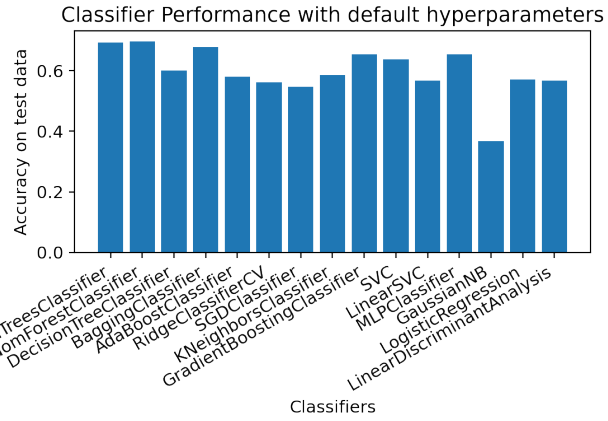


Fig. 7. Bar chart displaying accuracy of models without hyperparameter fine tuning.

*Unless stated otherwise, all models were performed on the full 96 feature dataset.* As previously hypothesized, amongst the worst performing methods were all of the linear models. The linear methods had accuracies that lied between $54.5\%$ and $57.0\%$, with the exception of the Gaussian Naive-Bayes which performed especially poorly, with an accuracy of $36.7\%$. This extremely poor result could be due to the majority of features not following a normal distribution - thus resulting in poor accuracy results [14]. It is worth noting that the hyperparameter spaces for these methods were not explored, but in the case of most linear methods, there are very few hyperparameters to set, hence, one cannot expect the accuracy levels to increase substantially to that of a non-linear method. The methods we hypothesized that would have the most potential based on our findings from EDA and other data knowledge did in fact perform reasonably well, especially relative to the linear models. The support vector machine with a RBF kernel achieved an accuracy of $62.7\%$, while the 3rd degree polynomial kernel hand an accuracy of $63.6\%$. The feedforward neural network with a single hidden layer of 100 units achieved $65.4\%$ accuracy, which without hyperparameter tuning is excellent as there are many hyperparameters that can be tuned. The decision tree reached an accuracy of $60\%$ and by combining multiple decision trees to form a Random Forest, we saw the highest accuracy of $69.6\%$. A similar model known as Extra Trees classifier performed very similarly with an accuracy of $69.3\%$. For completeness, the K neighbours classifier had an accuracy of $58.5\%$ and the linear discriminant analysis reached $56.6\%$. It is important to consider that a lot of these algorithms are subject to a level of randomness and those with larger hyperparameter spaces have a lot of scope to improve. The models with an accuracy level of over $60\%$ were all deemed worthy candidates for further fine-tuning through automated hyperparameter tuning and potential further feature selection with a focus on the simpler models where necessary.

### *B. Support Vector Machine*

A support vector machine [18] is a binary classifier, which in this case, is extended to a multi-class classifier using a "one-versus-one" approach [19]. Analysis will describe binary classification but it is within the context of the "one-versus-one" scheme. The intuition behind a SVM is to transform data into a higher dimensional space and establish a soft margin decision boundary, which is a hyperplane, that splits the dataset into two classes [16]. The defining hyperparameter of a support vector machine is the kernel function. Any of a linear, polynomial or a radial basis function can be used that decide how the data is transformed. A polynomial kernel of degree $d$ transforms data into a $d$ dimensional space and an RBF kernel, in a sense, transforms data into infinite dimensions, essentially creating a weighted nearest neighbour model that can perfectly divide and classify training data [19]. This makes an RBF kernel especially prone to overfitting (high variance) and highlights the importance of regularisation.

In terms of hyperparameters, as previously stated, the kernel function used heavily influences the sensitivity of an SVM model. This is important in the context of the dataset in question, as the aim is to model the complex underlying relationships, whilst not modelling the levels of noise found (i.e. overfitting the data), so an appropriate kernel function is key. There are two hyperparameters that aim to help reduce variance and prevent overfitting with an RBF kernel. These are the regularisation term, $C$, that sets the decision boundary margin size, and the kernel coefficient, gamma, which establishes the amount of influence that one training example will have on the model. As the regularization term increases, the less likely that overfitting might occur and vice versa. The higher the kernel coefficient, is, the more likely overfitting is to occur.

An exhaustive Grid Search Cross Validation [13] hyperparameter test was performed on the scaled dataset '$C$' values between :[1,10,100,1000], 'gamma':[0.001,0.01,0.1,1,10], 'degree':[1,2,3,4,5,10,20,50] and 'kernel':['poly','rbf'] were tested with the best parameters being '$C$': 10, 'gamma': 0.1 and 'kernel': 'rbf' with a mean 3-fold cross-validation accuracy score of $65.0\%$. This suggests that compared to the original model without hyperparameter tuning, it was deemed necessary to regularise the SVM more by increasing $C$ to 10 times the default value to prevent overfitting. However, a kernel function as complex and sensitive as the RBF kernel was needed to learn the complexity of the data. The classification accuracy obtained from the tuned model on the scaled data, with no feature selection, was $65.2\%$, a $2.5$ percentage point improvement from the original model without hyperparameter tuning. This is a reasonably solid score that demonstrates how there is noise in the data that has been accounted for by the regularisation term.

The classification report seen in Figure 9 for the final SVM model suggests a solid job of predicting food insecurity level. With the mapping defined in the Data Preparation section, the model's performance for each category can be evaluated. The weighted average F1-score of $0.64$ demonstrates similarly solid

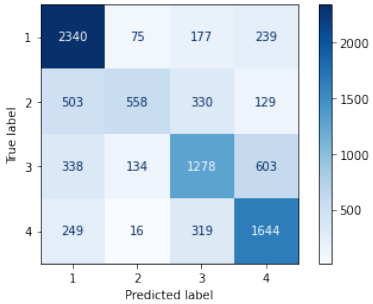


Fig. 8. Confusion matrix for the Support Vector Machine model.

precision and recall, demonstrating that overall, the model's precision and recall is reasonably strong (with a consistent weighted average 65% for each). Looking at individual class labels, it is evident that the SVM's strengths lie in predicting the two extreme classes, with the two highest class F1-scores of 0.75 and 0.68 belonging to Food Secure and Severely Food Insecure, respectively. A clear source of error of the model is it's weak performance with the Mildly Food Insecure (FI) true labels. This can be seen in the confusion matrix of Figure 8 and the classification report with an F1-score of 0.48 for Mildly FI and specifically a recall of 0.37. It's evident that the model struggles to predict Mildly FI data points well, with over a third of these data points being predicted as Food Secure.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.68 | 0.83 | 0.75 | 2831 |
| 2 | 0.71 | 0.37 | 0.48 | 1520 |
| 3 | 0.61 | 0.54 | 0.57 | 2353 |
| 4 | 0.63 | 0.74 | 0.68 | 2228 |
| accuracy | | | 0.65 | 8932 |
| macro avg | 0.66 | 0.62 | 0.62 | 8932 |
| weighted avg | 0.65 | 0.65 | 0.64 | 8932 |

Fig. 9. Classification report for Support Vector Machine.

### C. Feedforward Neural Network

A feed-forward neural network is simply a neural network where connections do not have feedback loops. The basic architecture of feed-forward neural network consists of an input layer, an output layer that corresponds to each label, and at least one hidden layer in between the two. Each layer of neurons following the input layer act as a function of weights and a bias of the layer before it, with the class that corresponds to the most activated neuron in the output layer being the prediction [19].

A given neural-network is trained with a training algorithm that aims to minimise a cost function during each training step. The resulting function born from a trained neural-network, depending on the architecture, can have 1000s of parameters and therefore be complex enough to model non-linear relationships within a high-dimensional dataset like the one being investigated. The sheer amount of parameters can lead to huge overfitting risk[19]. Training a NN over excessive epochs also presents a further overfitting risk. There are two main methods of preventing overfitting: Dropout rate in layers and kernel regularisation in a given layer. Intuitively, dropout aims to randomly "de-activate" a proportion of neurons in each layer, during every stage, to force each neuron to learn and prevent specific neurons from becoming overly important (or weights getting too large- which is often a signal of overfitting). Kernel regularisation is essentially per-layer regularisation that applies a penalty on a layer's kernel to penalise large weights and essentially attempt to prevent overfitting[19].

Keras [20] was used to train the neural-networks, while different neural-network architectures were trialled. The hyperparameters tested were: different hidden-layer arrangement (ranging from 2-5 hidden layers containing any combination of power of 2s of nodes up to 512), dropout rates of 0.4 to 0.8 and finally the learning algorithm, testing 'Adam' and SGD. The best performer based on accuracy score on the testing data was the NN that used the 'Adam' solver, with a hidden layer architecture of 256, 256, 128,128, 128 trained for 100 epochs on a scaled dataset reduced to the 70 best features (feature selection was used here to save time). We see from Figure 10 that the model has not overfitted the training set and has reached a strong level of accuracy of 65%.

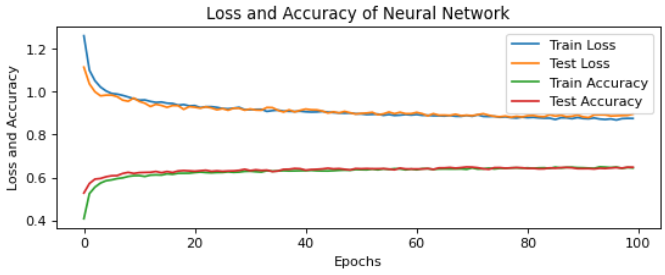


Fig. 10. Plot displaying the accuracy and loss of the training and testing sets of the neural network with optimal hyperparameters.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.73 | 0.81 | 0.77 | 3019 |
| 2 | 0.63 | 0.28 | 0.39 | 1403 |
| 3 | 0.54 | 0.65 | 0.59 | 2264 |
| 4 | 0.67 | 0.67 | 0.67 | 2246 |
| accuracy | | | 0.65 | 8932 |
| macro avg | 0.64 | 0.60 | 0.61 | 8932 |
| weighted avg | 0.65 | 0.65 | 0.64 | 8932 |

Fig. 11. Classification report for the Feedforward Neural Network.

The classification report for the final Neural-Network model suggests that once again, it did a solid job of predicting food insecurity on unseen data and generally performed very similarly to the SVM especially in terms of performance on the different classes. It predicted the extremes with similar F1-scores, but again, a clear source of error for the model is

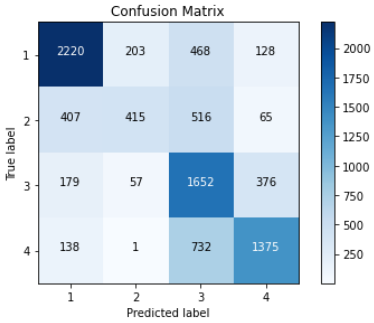


Fig. 12. Confusion matrix for the Feedforward Neural Network.

it's weak performance with the Mildly Food Insecure (FI) true labels and this can be seen on the confusion matrix and the classification report with an F1-score of 0.39 for Mildly FI and specifically a recall of 0.28. This is even significantly worse than the SVM which was already poor. It is evident that the model struggles to predict truly Mildly FI. This is reinforced by the confusion matrix seen in Figure 12 where more Mildy FI datapoints are actually predicted to be Moderately FI than Mildly FI.

### *D. Random Forest*

A decision tree is a type of non-parametric model that underpins many popular tree-based methods used in machine learning [16]. To understand a random forest, one must first understand a decision tree. In simple terms, a decision tree aims to divide a data set into small groups, to make each one as homogeneous as possible. This is done by repeatedly splitting the data (each split representing a node in the tree), attempting to minimise the Gini score of the resulting nodes i.e. the level of impurity [14]. This repeated sectioning of a given dataset allows decision trees to accurately learn non-linearly separable training data effectively, but the natural high-variance resulting from the methodology often results in overfitting, which, since it has been established that the dataset in this context is so noisy, is a problem. This is demonstrated by the accuracy score of 60.0% achieved by the decision tree in the exploratory modelling phase.

A popular way to build on the strengths and compensate for the weaknesses of a decision tree is to use what's known as a random forest classifier. A random forest aims to reduce the variance that results from a single decision tree by limiting each tree to potentially only a sample of the dataset and potentially a selection of the total features (at random). The aim of this is to cancel out error and reduce variance to prevent overfitting. Initial modelling found that this lead to huge improvement with the random forest improving on the score of the single decision tree by 9.6 percentage points, reaching 69.6% accuracy. This was our best original classifier.

The random forest is a relatively simple model. The underlying decision trees are somewhat complex, but besides tree hyperparameters, the number of estimators (trees) are particularly important because it is that diverse population of trees that is supposed to help reduce the variance of the model. For completeness, a GridSearchCV hyperparameter test was performed on the scaled dataset testing the following set of hyper parameters: 'bootstrap': [True, False], 'max_depth': [10, 50, 100, None], 'max_features': ['auto', 'sqrt'], 'min_samples_leaf': [1, 2, 4], 'n_estimators': [100, 500, 1000]. The best performing combination based on 3-fold cross-validation with a mean accuracy score of 70.2% was 'bootstrap': False, 'max_depth': 50, 'max_features': 'auto', 'min_samples_leaf': 1, 'n_estimators': 500. This indicates that limiting tree depth but not having a minimum number of leaf samples as well as a hefty 500 estimators allows for the most generalised model.

The final random-forest classifier using the best parameters from tuning was trained on the scaled dataset, again with no feature selection, as random forests naturally do perform feature selection. The final classification accuracy achieved on the unseen test data was 70.5% which is by far the best score and almost 1 percentage point higher than the random-forest that used default hyperparameters. F1 scores can be seen in Figure 13 as well as the confusion matrix in Figure 14.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.73 | 0.85 | 0.79 | 2831 |
| 2 | 0.76 | 0.42 | 0.55 | 1520 |
| 3 | 0.66 | 0.62 | 0.64 | 2353 |
| 4 | 0.68 | 0.80 | 0.74 | 2228 |
| accuracy | | | 0.70 | 8932 |
| macro avg | 0.71 | 0.67 | 0.68 | 8932 |
| weighted avg | 0.71 | 0.70 | 0.69 | 8932 |

Fig. 13. Classification report for the Random Forest.

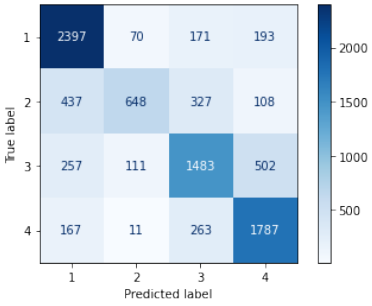


Fig. 14. Confusion matrix for the final Random Forest model.

The classification report for the final Random Forest shows a significantly improved all-round performance on virtually all of the classification metrics listed, compared to the SVM and NN. The accuracy score is the best we have seen so far with 71% accuracy. Furthermore, the weighted average F1-score of 70% is the highest achieved so far. Each class boasted a stronger F1-score than the previous methods, with scores for Mildly FI, Moderately FI and Severely FI improving by at least 0.05 compared to either of the previous models. We still see a

particularly poor recall with Mildly FI in the model but the improved score in the report and the higher concentration into the right cell on the confusion matrix indicate that the random forest doesn't do as poor a job with mildly FI as the other models. It is by far the best model developed. This is likely due to the robustness of random forests and how they're much less likely to overfit than and SVM or a NN. Adding estimators only adds more diversity, arguably reducing variance. This is important when dealing with such a noisy dataset.

### *E. Interpreting the Models Using Shapley Plots and Values*

A major criticism of advanced ML methods like ANNs, SVMs and Random Forests is that their complexity hurts interpretability (hence the term black-box methods) and makes it difficult to learn what relationships are contributing to predictions. We took advantage of Shapley values to help overcome this difficulty and understand which features were affecting the model output the most, obtaining very interesting results. As our second objective is to see which features are most impactful, we will only analyze the plots of the best model i.e. the Random Forest but in fact, the 3 main models tended to be most informed by a similar combination of features.

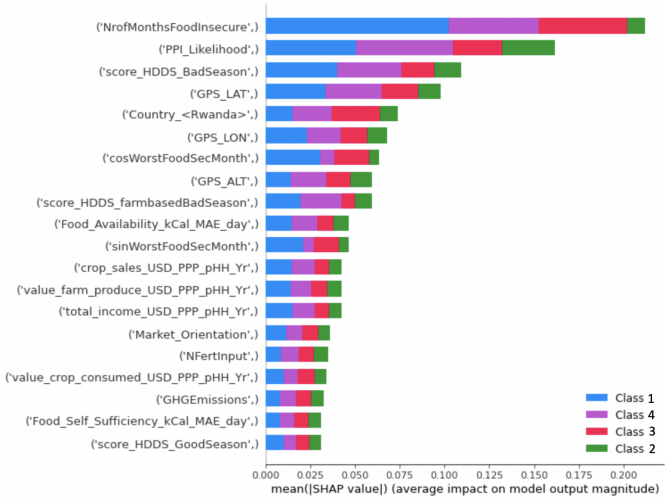


Fig. 15. Shapley bar plot displaying the most important features from the random forest.

A Shapley value is the the contribution of a feature value to the difference between the actual prediction and the mean prediction value [17]. Figure 15 shows that in the Random Forest model (not shown but also in other 2 models), the features that had the highest average impact on model output were number of months food insecure, which isn't surprising, and poverty probability index (PPI) likelihood, which is the surprisingly the second most impactful feature in the results. Figure 16 is a more detailed plot that demonstrates the relationship of different variables with the class of Severely Food Insecure. Red points indicate a high feature value, while blue a low feature value, with a positive SHAP value suggesting that a given point has "pushed" the model towards the Severely Food Insecure class. A negative SHAP value

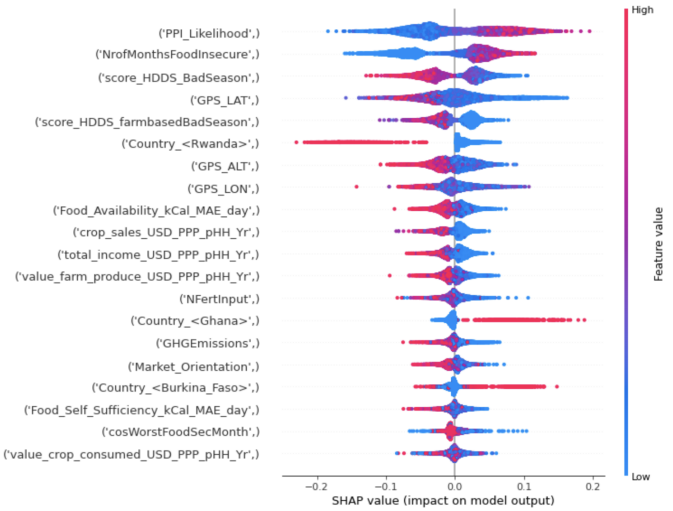


Fig. 16. Shapley scatter plot displaying the most important features from the random forest.

suggesting that a given point has "pulled" the model away from the Severely Food Insecure class. It is evident from Figure 16 that PPI likelihood and the number of months food insecure are positively related with being in the Severely FI class and interestingly. There is also a trend where low Longitude, Latitude or Altitude seems to positively impact being Severely FI.

Figure 15 also confirms some of the findings of the feature selection. After the top 10 features, one sees marginal differences between the average impacts of the features. This further suggests that actually, these models are relying on small contributions from many of the features to squeeze out as much predictive accuracy as possible and indicates that actually there is no one metric that is appropriate to predict Food Insecurity.

### *F. Further Evaluation and comparison of the Support Vector Machine, Neural Network and Random Forest models*

Clearly, the best model was the final random forest classifier developed, with superior accuracy and F1-scores all round. The final NN and SVM in comparison still performed reasonably well and were remarkably similar to one another. The models seemed to share many strengths and weaknesses. Firstly, the models seemed to have the best precision, recall and F1-scores for Food Secure and Severely Food Insecure (FI) whilst consistently performing the most poorly on the Mildly FI class. This is due to two factors. Firstly, the dataset, in terms of the distribution of the target variable 'Food Insecurity Level', is heavily imbalanced. Before imputation, the dataset contained around $10,000$ Food Secure observations, $9,000$ Severely FI observations, $6,100$ Moderately FI observations and only 3000 Mildly FI observations. This suggests that the models suffered from not having enough points to learn from for the Mildly FI class in particular and means that if predictive performance for all levels of FI are to be improved, the easiest way is to continually collect more data, as it seems that over $8,000$ values for a given label leads to really strong performance. This

is encouraging because assuming more surveys are conducted through RHoMiS, then over time, these models will continue to yield improved results as the dataset grows. It is also worth noting that the Mildly FI class had the highest proportion of imputed data (around $50\%$) and with that comes a natural level of uncertainty which is embedded within the models.

It is worth putting individual class performance into perspective as well. As previously stated, a model that can accurately predict FI would be useful in the context of an early warning system or in a system that might decide where to direct aid. This means that the priority is to accurately detect as many cases of Severe and Moderate FI in particular as possible (although wrongly classifying those who are Secure as Severely Insecure can result in misdirected efforts too). The 3 models looked at in detail managed to do this well. A recall score for the Severely FI class of $0.80$ for the final Random Forest is extremely important because it means that for a given case of somebody being Severely FI, the Random Forest will detect that with an accuracy of $0.8$. This would be very appropriate for an early warning system where in reality, the priority is to predict as many true positives as possible. A few false positives don't pose a major problem.

We also showed that the poverty probability index (PPI) likelihood, an asset based indicator, is significantly more informative about food insecurity than most traditional features such as farm income or total income. This is extremely significant, as one if an aid direction decision is being made by assessing just one measure, there is a strong argument now for that to be PPI likelihood.

## VI. Conclusion

In conclusion, machine learning algorithms, specifically Random Forest, Neural Networks and Support Vector Machines are excellent candidates in accurately predicting Food Insecurity given a noisy dataset. It seems the complexity and sensitivity afforded by these algorithms, paired with overfitting prevention methods such as regularisation and reducing variance by increasing estimator diversity provide highly effective methods to solve this problem. Accuracy scores of 65-70% reinforce it is possible to accurately classify Food Insecurity levels. Furthermore, as the dataset grows, it is likely these models will only perform better as it was evident from the detailed classification reports that the models weren't able to learn the classes with fewer data points as well, but performed strongly on the classes that had the most data. The strong recall performance on the class representing Severe Food Insecurity is encouraging because in most contexts, that will be of the highest priority, since it is those who are Severely Food Insecure who are most at danger.

Moreover, combining our models with Shapley allows us to predict more accurately than other works that would have used logistic regression whilst also retaining interpetability. Shapley plots allow us to understand even the more convoluted models more deeply and provide insights into what features best impact food insecurity predictions. Through this, we further confirmed that there isn't an individual dominant feature; the models rely on the combination of features to produce reasonable predictions. It is not as simple as looking at income. These findings might allow stakeholders in this domain to re-think what features contribute to food insecurity and potentially change their approaches.

## References


[1] John Holmes. *Losing 25,000 to Hunger Every Day*. 2021. URL: https://www.un.org/en/chronicle/article/losing-25000-hunger-every-day.

[2] Feeding America. *How do you measure hunger?* 2021. URL: https://www.feedingamerica.org/hunger-in-america/food-insecurity.

[3] *THE 17 GOALS — Sustainable Development*. 2021. URL: https://sdgs.un.org/goals.

[4] *The Rural Household Multi-Indicator Survey*. 2021. URL: https://www.rhomis.org/.

[5] Calogero Carletto, Dean Jolliffe, and Raka Banerjee. "From Tragedy to Renaissance: Improving Agricultural Data for Better Policies". In: *The Journal of Development Studies* 51.2 (2015), pp. 133–148. DOI: 10.1080/00220388.2014.968140. URL: https://doi.org/10.1080/00220388.2014.968140.

[6] Romain Frelat et al. "Drivers of household food availability in sub-Saharan Africa based on big data from small farms". In: *Proceedings of the National Academy of Sciences* 113.2 (2015), pp. 458–463. DOI: 10.1073/pnas.1518384112.

[7] Mark van Wijk et al. "The Rural Household Multiple Indicator Survey, data from 13,310 farm households in 21 countries". In: *Scientific Data* 7.1 (Feb. 2020). DOI: 10.1038/s41597-020-0388-8.

[8] William Nafack et al. Mar. 2021. URL: https://github.com/willianck/ADS.

[9] Leroy Wolins, Benjamin D. Wright, and Georg Rasch. "Probabilistic Models for some Intelligence and Attainment Tests." In: *Journal of the American Statistical Association* 77.377 (1982), p. 220. DOI: 10.2307/2287805.

[10] Jennifer Coates, Anne Swindale, and Paula Bilinsky. "Household Food Insecurity Access Scale (HFIAS) for Measurement of Food Access: Indicator Guide: Version 3". In: *PsycEXTRA Dataset* (Aug. 2007). DOI: 10.1037/e576842013-001.

[11] John Renze. *Outlier*. Apr. 2021. URL: https://mathworld.wolfram.com/Outlier.html.

[12] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations inR". In: *Journal of Statistical Software* 45.3 (2011). DOI: 10.18637/jss.v045.i03.

[13] *Sklearn Documentation*. 2021. URL: https://scikit-learn.org/stable/.

[14] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[15] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[17] Scott Lundberg. *SHAP*. 2018. URL: https://christophm.github.io/interpretable-ml-book/shap.html.

[18] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92* (1992). DOI: 10.1145/130385.130401.

[19] Christopher M Bishop. *Pattern Recognition and Machine Learning*. https://www.microsoft.com/en-us/research/people/cmbishop/downloads/. Springer, 2006.

[20] *Keras Documentation*. 2021. URL: https://keras.io/api/.