Robust Machine Learning for Food Security Forecasting

Master Thesis Presented to the
Department of Economics at the
Rheinische Friedrich-Wilhelms-Universität Bonn

In Partial Fulfillment of the Requirements for the Degree of
Master of Science (M.Sc.)

Supervisor:
Prof. Dr. Clara Brandi

Submitted in August 2023 by:
Gewei Cao
Matriculation Number: 3461232

Contents

Abstract

The food insecurity issue is serious in eastern Africa, facing the shocks such as the Covid pandemic and war in Ukraine, a robust machine learning is needed for food insecurity forecasting. This study combines the Uganda National Households Survey data and other open-source data to predict household food insecurity before and during Covid. It is shown that models based on decision trees behave robustly when facing the shock of Covid. These tree-based models provide flexibility for policymakers to trade off the cost and benefit of food insecurity aiding. We also find that demographic and asset features provide the most prediction power. Finally, tree-based methods are robust against limited features or un-updated training data when facing a shock, implying they are robust in practical scenarios.

1 Introduction

"Zart wäre einzig das Gröbste: daß keiner mehr hungern soll" —*Theodor W. Adorno*

In 2015, United Nations established Sustainable Development Goal 2 (SDG2), aiming to create a world free of hunger by 2030. However, according to Department of Economic and Social Affairs (2022) in 2019 there were still 559 to 650 million persons worldwide suffering from hunger, and this number increased by 161 million in 2020. Moreover, recent crises have further escalated the problem. With the breakout of Covid-19, WFP, UNICEF et al. (2022) estimated that there will be 78 million more undernourished people in 2030 than in a scenario in which the pandemic had not occurred. The breakout of Covid is unpredictable and has catastrophic aftermath. Such shocks usually cannot be considered in regular policies and plans. Meanwhile, there are many other ongoing shocks and crises like the war in Ukraine and natural disasters caused by climate change. Therefore, we have to consider shocks and sudden crises in food security studies now. The shock of Covid gives us an opportunity to study the influence and alleviation of shocks, and it reminds us to prepare for sudden crises in the future. [1]

Predicting the occurrence of a food crisis or food insecurity helps policymakers and international organizations to mitigate or prevent such negative events. Instead of causal inference, prediction is thus highly relevant and arguably important in the food insecurity context (Kleinberg et al. (2015)). With the development of machine learning (ML) and big data, starting with the study of Okori and Obua (2011), such data-driven technique implementations are emerging in this field (e.g. Lentz et al. (2019), Andree et al. (2020), Browne et al. (2021)). However, the previous machine learning studies on food insecurity rarely consider the shock and heterogeneity in the data, such as the shock of

[1] only sudden shocks and crises are the target of this study, long-onset crises such as sea level rise are not the target of us

Covid-19 and the war in Ukraine, especially at a time-series and micro level. For example, the study of Zhou et al. (2022) mentioned that they did not take sudden shocks into particular consideration, but recommended using human-based prediction to predict rare or unanticipated events. Barbosa and Nelson (2016) used a cross-sectional study to study the shock of a drought. Andree et al. (2020) and Andree (2022) considered the shock at a worldwide macro scale. Therefore, the robustness of the ML model for food insecurity prediction needs to be considered and assessed, especially at a micro-scale.

This study will focus on using ML to predict food insecurity in the Republic of Uganda, given the data and context of the breakout of Covid-19, trying to find out whether there exists a relatively robust ML algorithm when facing such shock and trying to find a high-performance ML model for the predicting task of food insecurity. Uganda is located in eastern Africa and is bothered by food insecurity, the study on such a typical eastern African country thus could be extended to other similar countries easily. Regarding our data sources, the Uganda National Households Survey data (UNHS) and other open-source data will be included, to provide the food security indicator and predictors. Regarding ML models used in this study, the benchmark algorithm is Logistic Regression, which is familiar to most economists. The famous Support Vector Machine (SVM) and other tree-based algorithms will also be used. These algorithms will be introduced in section 2.

This section is the introduction of the background and the motivation of this study. The next section will introduce the basic concepts of machine learning, the working flow, and the algorithms included in this study. Section 3 is a short literature review of the ML studies in food insecurity. Section 4 will introduce the food security situation in Uganda. Section 5 will introduce the data used, including the generation of food security indicator and the selected predictor variables. Then in section 6, two types of model designs are introduced, then a deep inspection of ML models' performance will be conducted. Finally, in section 7 a conclusion and discussion will be made.

2 Machine Learning

2.1 Introduction

Machine learning manages to fit complex data without simply overfitting and is able to give good out-of-sample predictions (Mullainathan and Spiess (2017)), and statistical learning theory provides the theoretical basis for many machine learning algorithms (Von Luxburg and Schölkopf (2011)). The key idea of machine learning in this context is to find general patterns for a given sample data and utilize such patterns for different purposes. If we want to make predictions on output data y for input data x, it is called supervised learning (e.g. linear regression), and if we only want to learn the structures and relationships for input data x but without output y, it is called unsupervised learning (e.g. Principal Component Analysis). In this paper, ML refers to supervised learning, i.e. we want to predict food insecurity y by using given data x.

Generally, ML solves problems of regression and classification. Regression here means the output variable y is a continuous real-valued variable, and classification means y is a categorical variable, usually y is binary. Most supervised ML algorithms need a measure of how well they are performing when fitting a model, the measure is called loss function, $L(\hat{f}(x), y)$, parameters or functions which can minimize the expectation of loss function $E_{(x,y)}(L(\hat{f}(x), y))$ are what supervised ML algorithms seek for. Note that assume there are p features, i.e. $x_i \in \mathbb{R}^p$ and X is a $N \times p$ matrix, where N is the number of observations. In the following sections, X will be called predictors or features. The name "feature" is usually used in a machine learning or computer science context.

As mentioned in section 1, this study will compare different ML algorithms and find the relatively robust one. The next subsection will introduce metrics used in classification problems. The regular working flow of ML will be explained in section 2.3, then in section 2.4 the ML algorithms used in this study will be introduced in detail. Finally, in section 2.5 an interpretable ML framework SHAP will be introduced, to help us find out important predictors in ML models.

2.2 Classification performance metrics

In this paper, food insecurity prediction is a classification problem, i.e. the response variable y is binary. Classification has advantages because one can construct multiple metrics to evaluate the performance of ML models and make policy implications. A food insecure observation is called a positive case and a food secure observation is called a negative case in the following table.

		True classes: Positive	True classes: Negative	Total
Predicted classes	Positive	TP	FP	P^*
	Negative	FN	TN	N^*
	Total	P	N	$P+N=P^*+N^*$

There are two types of important errors in the classification, Type I error (false positive) of inclusion (i.e., targeting those who are not food insecure) and Type II error (false negative) of exclusion (i.e. not targeting those who are insecure). Usually, type II error is more important in the food insecurity context. Some widely used metrics in classification are:

- Accuracy: $Accuracy = \frac{TP+TN}{P+N}$ totally correct prediction

- Precision: $Precision = \frac{TP}{P^*}$ correct prediction over predicted positive
- Recall (or Sensitivity): $Recall = \frac{TP}{P}$ correct prediction over true positive
- False positive rate: $FPR = \frac{FP}{N}$ false prediction over true negative
- F1 score: $F1 = \frac{2TP}{2TP+FP+FN}$ harmonic mean of the precision and recall

Moreover, the Receiver Operating Characteristic (ROC) curve is used to show the performance of a classification model at all classification thresholds. The ROC curve shows True Positive Rate (TPR) (or called recall) and False Positive Rate for different classification thresholds, this threshold is often a probability. The closer the curve to the top left corner, the better the model (see Figure 7 in section 6). Because this implies there exists a threshold with low FPR and high TPR. The Area Under Curve (AUC) is a good measure to compare such a pattern. The higher AUC, the more effective the model. A value of 1 for AUC means perfect fitting and a value of 0.5 or less than 0.5 for AUC implies the model performs the same as or worse than a random guess.

2.3 ML working flow

2.3.1 Data management

Data management is the first and one of the most important steps in any data analysis. This step contains the variable generation, missing data imputation, data merging, and feature selection (in machine learning jargon: Feature Engineering). Missing values in our data are imputed by the mode of the corresponding variable in the Uganda national scale. The feature selection step will be explained in detail in section 5. In addition, some open-source data have different district names, the district names in UNHS are taken as a benchmark, and those differences are matched manually or solved by a Uganda national scale mean imputation. After data management, 13,302 households are obtained for analysis.

2.3.2 Training & Testing data split

In ML, the data should be partitioned into three parts, training data, validation data, and testing data. The training data is used to train the ML model, the validation data is used for tuning hyper-parameters, the hyper-parameter tuning will be explained later in this section, and finally, the testing data is used to evaluate the performance of ML model. In this study, and usually, in ML, the validation data will not be created, considering the available data volume. Since the more data is used for training the ML model, the better the model's potential performance.

The UNHS data have a time dimension (date of interview), therefore, the time series information in the data should be considered. The training data should always be before the testing data, otherwise, the future data were used to predict the historical information, which has a risk of data leakage. The training and testing data are split monthly in this study.

2.3.3 Problem of imbalanced data

Food insecurity data here is imbalanced, the proportion of food severely insecure households is less than 5%. When data is imbalanced, a naive ML algorithm tends to treat the minority class as noise and ignores their influence. For example, if there are 100 households, and only 1 of them is food insecure, then if one makes a naive guess that any given household is food secure, a 99% accuracy will still be obtained.

To avoid the above problem, there are at least two potential solutions. Firstly, one can use weighted classes in ML model training, most ML packages contain the sample weight option, and in the Python sklearn package, the default balancing weight is inversely proportional to the number of observations for each class. This weighted response data will make the ML model more sensitive to the minority class. This study will choose this scheme to overcome the imbalance of data.

Secondly, one can use resampling techniques to generate synthetic samples for the minority class. Two techniques will be implemented, and both are oversampling techniques, i.e. generating more synthetic minority class observations. The first is the Synthetic Minority Oversampling Technique (SMOTE), which create new synthetic data by interpolation between several positive instances that lie together (Fernández et al. (2018)). The second is the Adaptive Synthetic Sampling Approach (ADASYN), which is based on the idea of adaptively generating minority examples according to their distributions (Fernández et al. (2018)).

However, the resampling techniques are threatened by the curse of dimensionality. One of the manifestations of the curse of dimensionality says, that for high-dimensional data the sampling density is proportional to $N^{1/p}$, where p is the dimension of predictors and N is the number of observations, for example, if $N_1 = 100$ means dense for a one-dimensional feature input, then in order to keep the same sample density, the sample size has to be $N_{10} = 100^{10}$ with 10 predictors (Hastie et al. (2009)). Therefore, before resampling, the dimensionality of input features has to be reduced. The ML result section will show the resampling performances.

2.3.4 Hyper-parameter tuning

In the ML model, usually, there are some exogenous parameters, these parameters are not estimated during model training, but are given before model fitting, these parameters are called hyper-parameter in the context of machine learning. Hyper-parameters are usually

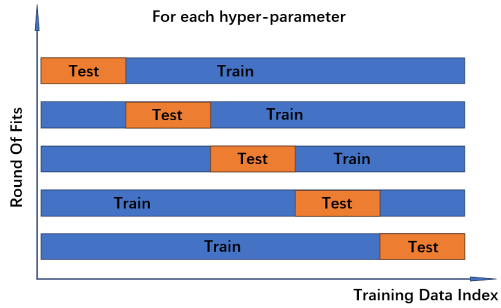


Figure 1: Visualization of Cross Validation

estimated by cross-validation. Cross-validation partitions the training data into K folds, usually $K = 5$ or 10, this study uses $K = 5$. Given a series of hyper-parameters, for each value of the parameter, there will be K rounds of fit, and in each round one of K folds of data is selected as testing data, the left $K - 1$ folds data are training data, then we could obtain one performance metric value. In this paper, this metric is AUR ROC, because AUC can reflect the flexibility of the trade-off between accuracy and recall. Finally, K metric values are obtained, and the mean value is the estimated performance of the given hyper-parameter, then the best-performed parameter will be chosen as what will be used for training the ML model. Figure 1 is an illustration of how 5 folds cross-validation works, but the folds sampling is unnecessary with the order of data indices.

As mentioned above, UNHS data has time labels, but for the purpose of nowcasting (in other words, short-term forecasting), time-series cross-validation will not be used, instead the cross-sectional cross-validation is selected, i.e. for the training set, one does not consider any time label in the cross-validation as in Figure 1.

2.3.5 Performance evaluation

After training the model, testing data is used to evaluate the performance of the ML model. Testing data has never been included in the training and hyper-parameter tuning process, thus, it is "new" data to this model, which simulates the practice of ML in a real-world problem. In this study, the performances are evaluated by AUC ROC, since AUC ROC could reflect the trade-off and dynamics of the recall, precision and other metrics we are interested in. These trade-offs could help the policymaker understand and manage the humanitarian aiding projects more effectively and efficiently. A higher AUC implies a better absolute prediction performance and stronger flexibility for cost and efficiency trade-offs.

2.4 Machine learning algorithms in this study

2.4.1 Logistic Regression

Logistic regression is quite famous in econometrics, usually, it is solved via maximum likelihood estimation, and the loss function L_{log} is

$$L_{log} = -\sum_{i=1}^{N}[ln(1-p_i) + y_i ln(\frac{p_i}{1-p_i})]$$

where

$$p_i = \frac{e^{X_i\beta}}{1+e^{X_i\beta}}$$

If simply minimize the L_{log} and find the β vector, in such a high dimensional and complex food security prediction problem, there is a risk of over-fitting. Thus, ML uses regularization to mitigate such risk. Regularization controls the scale of parameters, and in the logistic regression case, L_1 penalty is added to L_{log}, i.e.

$$\min_{\beta} \; L_{log} + \lambda \sum_{j=1}^{P} |\beta_j|,$$

regression with L_1 penalty is called Lasso (least absolute shrinkage and selection operator) regression. In the Lasso regression case, regularization not only controls the scale of parameters but also selects a set of most significant features i.e. some β_j will be given 0 value. The tuning parameter (hyperparameter) λ controls the influence of the penalty term on the minimization problem, and it is given exogenously, usually chosen via cross-validation.

2.4.2 Support Vector Machine

Support Vector Machine (SVM) is a very famous ML algorithm. It provides a nice intuition for the classification problem. Data $x_i \in \mathbb{R}^p$, $i = 1, 2, ..., N$, and they have two different labels y, and in this case, $y \in \{-1, +1\}$. In this p dimensional hyper-space, SVM tries to find a hyperplane to separate data into two parts, $y = -1$ and $y = +1$. The "best" hyperplane which separates data is what the support vector classifier wants to find. If data is separable, the "best" hyperplane is the farthest separating hyperplane from all training observations, and the smallest distance of all training observations from the hyperplane is called *margin*. However, usually there is no such perfect hyperplane, and it is preferred that such a hyperplane be more flexible. Thus, soft margin is used, which allows some training points on the wrong side of the hyperplane or in the margin.

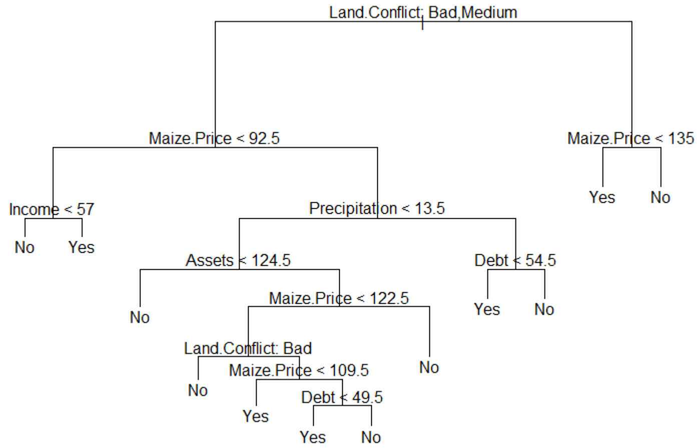


Figure 2: Example of a single decision tree

The optimization problem of the support vector classifier is

$$\max_{\beta_0,\beta_1,\ldots\beta_p,\epsilon_1,\epsilon_2,\ldots\epsilon_n,M} M$$
$$s.t. \sum_{j=1}^{p} \beta_j^2 = 1$$
$$y_i(\beta_0 + \boldsymbol{\beta} \cdot X_i) \geq M(1-\epsilon_i)$$
$$\epsilon_i \geq 0,\ and \sum_{i=1}^{N} \epsilon_i \leq C$$

, where C is non-negative tuning parameter, and ϵ_i are the slack variables, which allows x_i on the wrong side of margin or hyperplane. Solving the parameters $\boldsymbol{\beta}$ one can obtain a hyperplane $f(x) = \beta_0 + \sum_{j=1}^{p} \beta_j x$ which is called linear support vector classifier. Notably, for such a linear support vector classifier, the hyperplane only depends on training observations lying on the margin or violating the margin, and these observations are called support vectors (James et al. (2013)).

However, data usually cannot be separated linearly, and the extended version of the support vector classifier is the SVM. SVM uses the kernel $K(x_i, x_j)$ to enlarge the feature space of X, and one can solve the optimal classifier analytically for some kinds of kernels. The decision function of a kernelized SVM has the form $f(x) = \sum_{i=1}^{N} \alpha_i K(x, X_i) + b$, where X_i is the ith training observation, the representer theorem guarantees this.

2.4.3 Tree-based methods

A typical decision tree is intuitive, it partitions the feature space into a set of regions and then fits a simple model (like a constant) in each one (Hastie et al. (2009)). Figure 2 is an example of a decision tree. By doing this, a decision tree can reflect the non-linearity and complex interactions of our data. Those regions are called terminal nodes, and each terminal node can make a regional prediction, for instance, the majority class or the mean of the response variable for that terminal node. The split for each pair of nodes is made according to some form of loss function or some criteria, such as entropy or information gain in a classification context. There could be some stopping limits, otherwise, a tree can have many terminal nodes as observations. Meanwhile, to avoid over-fitting, a technique called tree pruning is introduced. Tree pruning first needs a fully grown tree T_0, then prune to control the number of terminal nodes.

However, the above decision tree algorithm is still non-robust and has high variance. Thus, techniques of bagging and boosting are introduced. Bagging uses totally B bootstrapped training samples and fits a tree for each sample, then average the predictions of these trees to make a prediction. Different from bagging, boosting trees do not need bootstrapped samples. It uses a weak learner, for instance, a tree (stump) only has one split and two terminal nodes. With such weak learners, for each fit, $\hat{f}(x)$ is only updated at a little learning rate λ, for example, $\lambda = 0.01$. Then fit B times, and obtain the boosted model as $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$. In other words, for each round of fitting, target data is modified, boosting fits the new "residual" in each round. Nevertheless, in the context of classification, instead of "residual" in the regression context, the $log(odds) = log(\frac{p}{1-p})$ is usually used in each round of fitting.

Random Forest

The random forests algorithm is a bagging tree-based method. Although a single decision tree is intuitive and can include high dimensional interactions and non-linearities of features, it could have a high variance for testing data. In random forests, with the bagging of trees, the variance of function is reduced, and trees are decorrelated.

To be specific, drawing a bootstrap sample of size N and using m out of p features (predictors) to grow a tree T_b, repeat this B times, then a set of trees $\{T_b\}_1^B$ is obtained. Then use the average prediction results of $\{T_b\}_1^B$ to make the final prediction, usually random forest takes the mean of single predictions in the regression case and makes a majority vote in the classification case.

XGBoost

eXtreme Gradient Boosting (XGBoost) is proposed by Chen and Guestrin (2016), it is a regularized gradient boosting method. XGBoost substitutes loss function $L(y_i, \hat{y}_i)$ by its second-order Taylor expansion $L(y_i, \hat{y}_i^b) = L(y_i, \hat{y}_i^{b-1} + f^b(x_i)) \approx L(y_i, \hat{y}_i^{b-1}) + L'(y_i, \hat{y}_i^{b-1}) f^b(x_i) + \frac{1}{2} L''(y_i, \hat{y}_i^{b-1}) f^b(x_i)^2$, where $f^b(x_i)$ is the predicted value of x_i for bth round fitting, and $\hat{y}_i^{b-1}$ is the fitted value for the former total $(b-1)$ rounds. Thus, $\hat{y}_i^{b-1}$ is constant in the bth round. Rewrite $L'(y_i, \hat{y}_i^{b-1})$ as g_i and $L''(y_i, \hat{y}_i^{b-1})$ as h_i. Meanwhile, for XGBoost decision trees, regularization for pruning and controlling the scale of predictions is given by $\Omega(f_b) = \gamma|T| + \frac{1}{2}\lambda\sum_{j=1}^{|T|} f_j^b(x)^2$. Then, rewrite the optimization problem in round b as

$$\min_{(f_1^b(x), f_2^b(x), \ldots, f_T^b(x))} \sum_{j=1}^{|T|}(f_j^b(x)G_j + \frac{1}{2}f_j^b(x)^2(\lambda + H_j)) + \gamma|T|,$$

where $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$. I_j is the set of all observations that belong to the jth terminal node, and there are T terminal nodes. In addition, remember that $f_j^b(x)$ is the predicted value of terminal node j which is constant for all $x \in I_j$.

For a given structure of a tree, compute the predicted value for terminal node j:

$$f_j^b(x) = -\frac{G_j}{H_j + \lambda}$$

In addition, the splits for a single tree are made according to the relationships of $\frac{G_j^2}{H_j+\lambda}$ and γ. How the splits are made will determine the structure of a tree.

2.5 SHAP

In ML, model interpretability is usually an issue. Unlike the linear regression model or logistic regression model, the ML models are very complex and very hard to explain. Therefore, the Shapley values are introduced. Shapley value is used to evaluate the individual contribution in cooperative game theory firstly proposed by Shapley et al. (1953). Shapley value for feature j at value x_i is calculated by:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!}[f_{S \cup \{j\}}(x_{i,S \cup \{j\}}) - f_S(x_{i,S})]$$

, where F is the set of all features, and S is one of all subsets of F. For the target feature j, $f_{S\cup\{j\}}(x_{i,S\cup\{j\}})$ is a model trained by features $S \cup \{j\}$, i.e. $f_{S\cup\{j\}}(x_{i,S\cup\{j\}})$ is the prediction for feature values in set $S \cup \{j\}$ that are marginalized over features that are not included in set $S \cup \{j\}$, so as $f_S(x_{i,S})$. Then, $f_{S\cup\{j\}}(x_{i,S\cup\{j\}}) - f_S(x_{i,S})$ can represent the effect of feature j. The term $\frac{|S|!(|F|-|S|-1)!}{|F|!}$ is a weight factor, that represents the probability of feature set S is selected (consider the permutation). Thus, ϕ_j could represent the contribution of feature j for observation x_i.

However, the calculation of the above Shapley value is costly when the feature dimension p is large. Thus, SHAP (SHapley Additive exPlanations) proposed by Lundberg and Lee (2017) is introduced. SHAP can solve Shapley value by an additive feature attribution method. For a given observation x_i, when approximating the model $\hat{f}(x_i)$ locally, x' is a simplified version of x_i, $x_i = h_x(x')$, local methods try to ensure $g(z') \approx \hat{f}(h_x(z'))$ whenever $z' \approx x'$. Meanwhile, $z' \in \{0,1\}^M$, where 0 represents the absence of the corresponding feature, and 1 represents presence, $M = p$ in our study. $\hat{f}(x_i)$ could be rewritten as

$$\hat{f}(x_i) = g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j$$

where $z'_j = 1$, $\forall j$. ϕ_j is the Shapley value of feature x_j of observation x_i. In addition, ϕ_0 is the expected value of model $\hat{f_X}$. SHAP could solve the Shapley value of most ML models.

SHAP enables researchers to interpret ML models, but not in a causality manner. In practice, Shapley value of all features for all observations will be computed, and this estimated Shapley value is used to evaluate the contribution of each feature in the trained ML model. Model interpretation, validation, and feature selection could be conducted with the help of SHAP.

3 Literature Review

ML implementation in food insecurity is an emerging field, which combines advanced big data and algorithms with existing food insecurity studies. Zhou et al. (2022) proposed principles for ML in the field of food security. They stated food security modeling should: (1) capture interactions and nonlinearities, (2) consider interpretability, (3) conduct error analysis, and when making choices researchers should ask themselves: (1) which outcome to predict, (2) how to target rare events, (3) how to evaluate the model. These principles are concise summaries of existing ML food insecurity studies.

ML always requires adequate data, and data availability has to be considered. Lentz et al. (2019) considered the cost of data collection and model performance, and achieved a recall of 0.83 and an accuracy of 0.76 in the food security prediction of Malawi. Open-source secondary data is helpful in these studies, for instance, in the study of Martini et al. (2022), their model performed best when including both secondary data and previous food security assessments. But they also show that only using secondary data also could generate a satisfactory result, this implies that their model can be implemented in regions where lack of previous food security index data. Meanwhile, Westerveld et al. (2021) used data from 2010 to 2018 in Ethiopia to predict food insecurity, and they utilized open data to make their methods can be implemented in other countries.

Previous studies also considered crises and shocks of food insecurity. At the micro level, Barbosa and Nelson (2016) used agricultural household survey data to predict the food security status given a drought shock. In the final model, it gives a 77% accuracy and 84% recall. They also confirmed the role of financial capital and agricultural-related features in assuring food security. At the macro level, Andree et al. (2020) used data from 21 developing countries from 2009 to 2020. They only used data for which no previous food insecurity occurs to train their model because their model focuses on predicting outbreaks of food insecurity. It turns out their model outperforms the prediction results provided by FEWS NET (Famine Early Warning Systems Network). Andree (2022) applied their model to a total of 151 countries from 1999 to 2019. In their section on future forecasting till 2027, the results were generated by projecting the covariates forward using the IMF's WEO outlook of April 2022. This enabled them to access the impact of the pandemic and the Ukraine war and to implement scenario analysis. Cooper et al. (2021) used microdata from 75 countries to build a global food insecurity forecast till 2030. The effect of Covid-2019 was taken into consideration. They made long-term predictions for the world food insecure population and found heterogeneous food insecurity patterns within different income-level countries.

ML is not only for prediction, it could also give researchers more information on the driving predictors, although not in a causality manner. Gao et al. (2020) used decision tree and random forest with the 2008 national survey data to predict food insecurity in Afghanistan. Their models have around 80% recall. Features such as income, household size and farm-related measures play a big role in prediction. Meerza, Meerza and Ahamed (2021) predicted food insecurity in Bangladesh, and household size, household expenditure, and livestock assets were the top three important predictors. Zhou et al. (2022) compared the importance of price, weather, assets and geographic features in different eastern African countries.

Finally, food insecurity prediction could be either a classification problem as in Barbosa and Nelson (2016) and Hossain, Mullally and Asadullah (2019), or a regression problem as in Zhou and Baylis (2019) and Cooper et al. (2021). Meanwhile, ML on food insecurity implementation is not limited to developing countries, for example, Nica-Avram et al. (2021) studied food insecurity in the UK, and achieved an accuracy and a recall of 0.75. Back to Uganda, Zhou et al. (2022) predicted food insecurity indicators using data from 2009 to 2011 in Uganda, and Okori and Obua (2011) predicted the famine in Uganda using data from July 2004 to June 2005. All of these above show the flexibility of ML and the complexity of food security issues.

4 The Republic of Uganda

The Republic of Uganda, total area of 241,038 km^2, situated in Eastern Africa, is a landlocked nation bordered by Kenya to the east, South Sudan to the north, the Democratic Republic of the Congo to the west, Rwanda to the southwest, and Tanzania to the south. A significant part of Lake Victoria, which is shared with Kenya and Tanzania, is located in the southern region of the country. With an average elevation of 900 meters above sea level, Uganda's eastern and western borders are marked by mountains. According to the

Uganda National Household Survey 2019/20 (UNHS), there are 40.9 million population in Uganda, and 49% of them are male. Meanwhile, 54% of the population in Uganda is below 18 years old. Within the working population, 68.1% of them work in the sector of agriculture, forestry and fishing.

Furthermore, Uganda's poverty rate is 20.3% in UNHS 2019/20, which was 18.7% in 2019, but after the breakout of the pandemic, it raised to 21.9%. In the round of the October/November 2022 Uganda-High-Frequency Phone Survey (HFPS), 56 % of the population was moderately food insecure and 15 % was severely food insecure. Thus, food insecurity in Uganda is severe, less than half population is able to access or buy enough food. Besides the shock of Covid-19, additionally, the ongoing conflict in Ukraine, which has disrupted global trade, poses further challenges to food security in Uganda. For instance, according to the FAOSTAT data, in 2020, Uganda's agricultural fertilizer use of the nutrient nitrogen, phosphate P2O5, and potash K2O is fully dependent on import, but the Ukraine war is disrupting the supply chain of fertilizer (WFP, UNICEF et al. (2022)).

In addition, in UNHS 2019/20, the average Dietary Energy Consumption (DEC) in Uganda is 2393 kcal/person/day, more specifically, before the pandemic, DEC was 2437 kcal/person/day, while during the pandemic, DEC decreased to 2359, this number varies among different regions as well. Notably, 47% households in urban areas are classified as food poor by the Uganda government, and it is 22% in rural areas. Specifically, on average, Staples (cereal, roots and tubers) are consumed on a daily basis while meat, fruit and milk products are the least consumed in a week, we can find such patterns in the data description section as well. Figure 3 shows the food insecurity ratio of the surveyed households (in %) in each district of Uganda, food insecurity threshold is FCS (Food Consumption Score) ≤ 21, in the later section we will explain the meaning of FCS. We can find that the northern and eastern parts of Uganda have higher ratio of food insecure households, which is somehow consistent with the finding of Okori and Obua (2011).

In conclusion, as an Eastern African country, Uganda has issues of poverty and faces the threat of significant food insecurity. Accurate predictions of food insecurity can assist the people of Uganda, as well as public sectors and international humanitarian organizations, in implementing more cost-effective strategies to achieve development goals, such as the Uganda Nutrition Action Plan or SDG2.

5 Data of Uganda

Uganda National Household Survey (UNHS) is undertaken by the Uganda Bureau of Statistics. Food security indicator (response variable) and some other predictors are

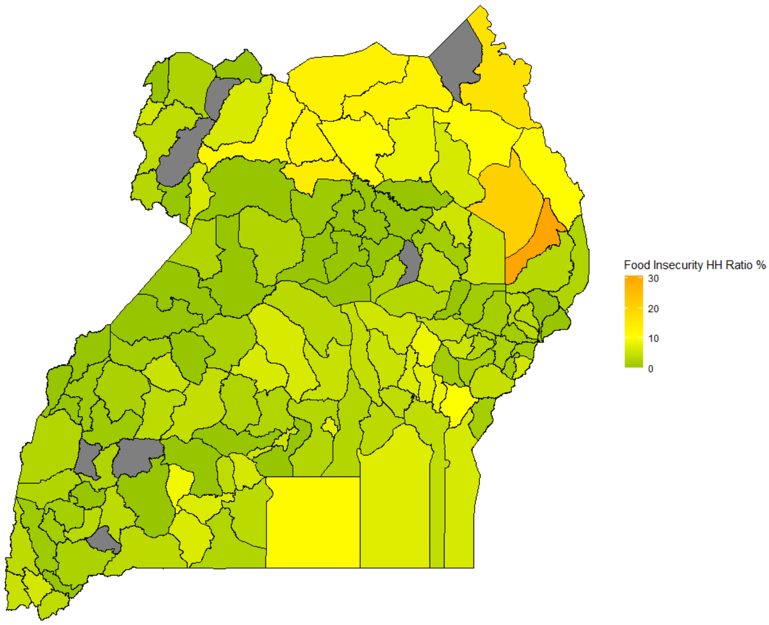


Figure 3: Uganda Food Insecurity Map of UNHS 2019/20

provided by UNHS. The UNHS of 2019/20 and 2016/17 data are used. The 2019/20 data covers from September 2019 to February 2020 and July 2020 to November 2020, a total of 11 months. After the data-cleaning process, some households will be dropped. Meanwhile, other kinds of open-sourced data will also be used. In addition, this paper focuses on machine learning and prediction for specific data, the population weight in the survey data will not be used, because in a descriptive study (e.g. predict the national scale of food insecure rate in Uganda), imputation and adjustments could be made.

5.1 Food security indicator

Considering the data availability, the Food Consumption Score (FCS) is chosen as the food security indicator in Uganda. The FCS is a composite score based on dietary diversity, food frequency, and relative nutritional importance of different food groups. FCS was first used in Southern Africa in 1996 and is frequently used in food insecurity ML prediction papers (e.g. Zhou et al. (2022),Lentz et al. (2019) and Martini et al. (2022)). FCS is calculated as $FCS = \Sigma_{i=1}^{9} x_i w_i$, where the x_i is the number of days in the past 7 days that a household consumed food item i, and w_i is the nutrition weight for i, i and w_i are shown in Table 1. Thus, $FCS \in [0, 112]$.

Food groups(i)	Weights (w_i)	Food groups(i)	Weights (w_i)
Main Staples	2	Meat and fish	4
Pulses	3	Dairy	4
Vegetables	1	Oil and Fat	0.5
Fruits	1	Sugar	0.5
Condiments	0		

Table 1: FCS Components (WFP (2008))

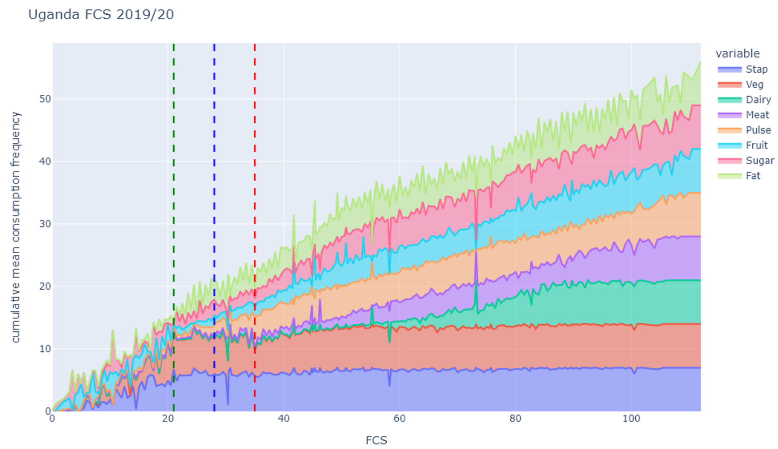


Figure 4: Cumulative mean consumption frequency 2019/20

FCS data in this paper is calculated with the UNHS data, rather than a standard FCS questionnaire. After calculating FCS, households are categorized into 3 classes, for those $FCS \leq 21$ are in the "Poor" diet group, for those $FCS \in (21, 35]$ are in the "Borderline" group, and for those $FCS > 35$ are in the "Acceptable" group. The threshold of 21 and 35 are usually used to indicate food consumption and security status. The value 21 comes from an expected daily consumption of staples and vegetables for the last 7 days, and 35 comes from an expected daily consumption of staples and vegetables complemented by a frequent (4-day/week) consumption of oil and pulses. Meanwhile, WFP (2008) suggested that the use of FCS and the threshold value need to be validated.

Figure 4 shows the mean food consumption frequencies for each value of FCS in Uganda 2019/20 UNSH, the green dashed line is the threshold of 21, the blue dashed line is 28, and the red dashed line is 35. 28 is an alternative threshold for "Poor" diet in WFP (2008), because in some cases, for low FCS households, the sugar and oil consumption may be daily, and combined with daily staple food consumption, FCS already reaches 21, but this 21 does not reflect the borderline diet. Therefore, 7 is added to 21 and 35. From Figure 4, we can find that neither the consumption of sugar nor fat/oil is not reaching 7 for low FCS groups, thus in this paper, we use 21 and 35 as the thresholds.

In addition, we need to evaluate if the created FCS is a good proxy for food security (WFP (2008)). The validation of created FCS score is shown in Table 2, and we can conclude that the created FCS can be used as a good indicator of food security. Note that for Subjective Income Stability, 1 means very unstable, 2 means somewhat stable, and 3 means stable.

In summary, the 2019/20 FCS of 13,302 households are calculated, and the FCS distribution of UNHS is shown in Figure 5. Figure 5 shows that most households are in the

Variable	Corr. Coef.	p-value
Subjective Income Stability	0.231 (Pearson)	0.000
Subjective Income Stability	0.235 (Spearman)	0.000
Total Expenditure on Food	0.605 (Pearson)	0.000
Proportion of Food (value) Received in Kind/Free	-0.155 (Pearson)	0.000

Table 2: Correlation Coefficients with FCS

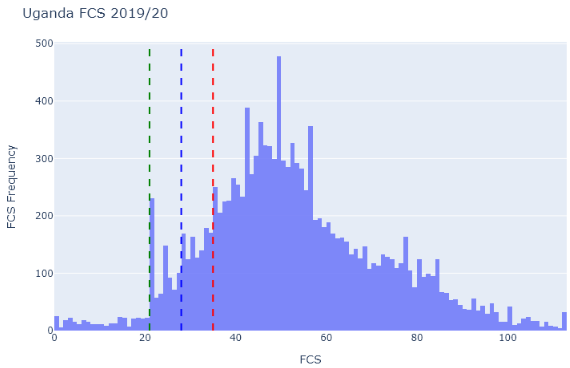


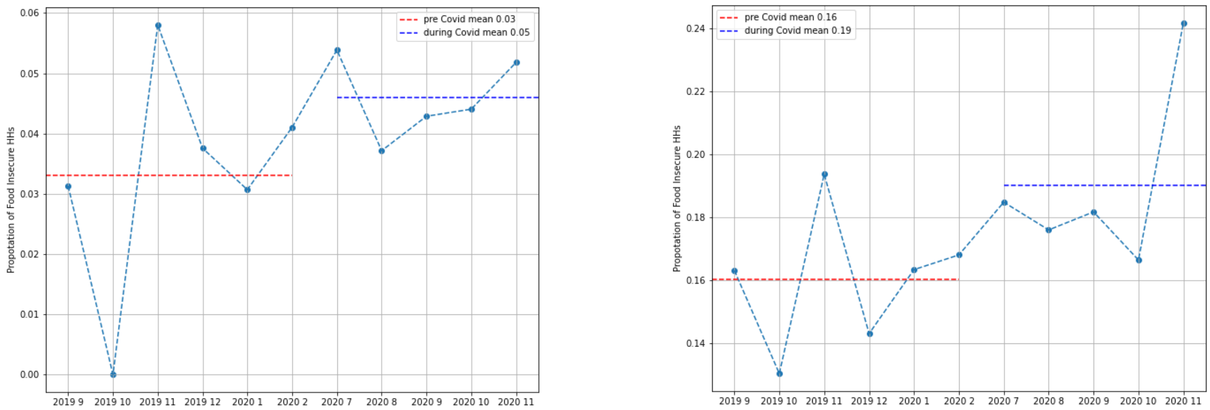
Figure 5: Distribution of FCS in Uganda 2019/20

"Acceptable" group. Notably, at the bar of the "Poor" threshold of 21 (right-hand side of the green dashed line), which is $FCS \in [21, 21.9]$, this high bar contains 230 observations and 214 of them are equal to 21 (in the "Poor" group). 4.16% of the households are in the "Poor" food security situation, and 17.72% of households are in the "Borderline" group. If we exclude those who have 0 FCS, then 3.99% households are in the "Poor" group, but in our study, the 0 FCS will be included.

Finally, Figure 6 shows that the proportions of food insecure households are higher during Covid than before Covid for both thresholds for food insecurity. Meanwhile, in Figure 6.b there is a periodic pattern in September, October, and November of 2019 and 2020, and the food insecure proportion during Covid is higher than before for the same months. Thus, we can conclude that Covid caused an increase in food insecurity, which is consistent with the UNHS report mentioned in section 4.

5.2 Predictors

The predictors used in this study consist of two dimensions, macro and micro predictors. The first is open-source macro data, using such as satellite nightlight data, climate data, conflict data, and Normalized Difference Vegetation Index, which are commonly used in other food insecurity studies. The second is micro household level data, which are provided by UNHS.



(a) "Poor" group

(b) "Poor" and "Borderline" group

Figure 6: Proportion of food insecure HHs

5.2.1 Open source data

Nightlight data

Nightlight data is used as a proxy of economic variables (e.g. Weidmann and Schutte (2017), Yeh et al. (2020)), especially in those developing countries, whose detailed economic data are not available. Because nightlight data is open-sourced, researchers can utilize it at a low cost. Meanwhile, it provides more information on local development in developing countries. This paper uses the nightlight data provided by the Earth Observation Group[2] (Elvidge et al. (2013)), and in order to be consistent with UNHS, the data of Uganda's subnational administrative boundaries (map of Uganda) provided by UN The Humanitarian Data Exchange[3] project is used to extract the mean nighttime illumination for a given region.

The nightlight data is extracted on the scale of District and County. Although the map data is provided by the Uganda Bureau of Statistics, a few names of administrative counties in both datasets do not match, and for those not matched, an imputation by the data of the corresponding District happens.

Climate data

The climate variables prepared for this paper are monthly sum precipitation (mm) and monthly 2 meters range above ground temperature (celsius degree). *These data were obtained from the NASA Langley Research Center (LaRC) POWER Project funded through the NASA Earth Science/Applied Science Program.*[4] They are extracted with the map data as the nightlight data on the district scale because there are 169 cells of climate data, but the number of counties is 194 in UNHS, thus, keep the temperature and precipitation variation in the district scale should be enough. In addition, another source of rainfall

[2]https://eogdata.mines.edu/products/vnl/
[3]https://data.humdata.org/
[4]https://power.larc.nasa.gov/data-access-viewer/

data is also used, it is provided by VAM Food Security Analysis[5], and for those districts without data, an imputation of Uganda national scale data happens.

Conflict data

The conflict data are obtained at *Armed Conflict Location & Event Data Project (ACLED),*[6] more could be found at Raleigh et al. (2010). There are totally six kinds of conflicts recorded in Uganda from 2016 to 2020: Strategic developments, Battles, Riots, Explosions/Remote violence, Violence against civilians, and Protests. Each type of conflict event is counted for a given district and date. Meanwhile, different from Martini et al. (2022) which used the time difference fatalities, The total fatalities for each month for all kinds of conflicts are generated, to reflect the fierceness of the conflict. For all seven generated variables, the missing values are replaced by the mode of that variable values for the covered two years.

NDVI data

NDVI (Normalized Difference Vegetation Index) is used as a predictor for food security in the ML context as well (e.g. Martini et al. (2022) and van der Heijden et al. (2018)). The NDVI used in this paper is downloaded from VAM Food Security Analysis[7]. For those districts without NDVI data, an imputation of Uganda national scale data happens. Lagged data and the mean of the past four months' data are added.

5.2.2 UNHS data

In UNHS data, households' demographic variables, FCS district distributional data of UNHS 2016/17, wealth indicators, income, and living standard indicators are used as predictors. Some categorical variables are needed to be transformed into different forms, such as one-hot encoding (dummy variable encoding) or frequency encoding (replacing categories by their frequency value). After checking the marginal distribution of some features on the food security status, some of the variables are log-transformed, to help the ML model become more sensitive against those features.

Only the district food insecure ratio of Uganda 2016/17 is chosen as a feature from the UNHS 2016/17 data, representing the historical food insecurity distribution. Because some microfeatures in UNHS 2019/20 do not appear in UNHS 16/17 data, it is impossible or inconvenient to generate a large training data which includes all 2016/17 features.

[5] https://dataviz.vam.wfp.org/seasonal_explorer/rainfall_vegetation/visualizations
[6] www.acleddata.com
[7] https://dataviz.vam.wfp.org/seasonal_explorer/rainfall_vegetation/visualizations

5.2.3 Final features

Combining macro open-sourced data and micro household data, time-invariable features such as asset and geographic features, and time-variable features such as meteorological and demographic features are included in our study. These features could cover shock-sensitive and insensitive variables, which give ML more power in food insecurity prediction.

After the data management and feature engineering, there are 50 features selected to train ML models. A detailed variable list could be found in the appendix. The correlation coefficients matrix of continuous variables is shown in the appendix, there is no strong correlation between any pairs of features. Some open-source macro-level data are also manipulated, such as using the mean value of the previous four months' data or their logarithm transformation as features.

6 ML Results on Food Insecurity Prediction

6.1 Introduction and model design

The goal of this study is to evaluate and find robust ML models in a nowcasting context. Nowcasting, or short-term forecasting here means the time labels and future predicting will not be considered, only the present data will be input into the ML model, and the output is the current food insecurity indicator. Nowcasting is important in the food security context because usually food insecurity is hard to evaluate, and its indicators are costly to generate. Typical ML studies on food security or poverty take nowcasting as a goal, such as Martini et al. (2022) and Browne et al. (2021).

There are multiple choices to design the prediction models in this food security context. To evaluate the robustness of ML models, either the shock of Covid in the training data or the shock in the testing data should be covered since it is assumed that in UNHS data there is a pattern difference between before and during Covid.

Firstly, an ex-post robust design is proposed, which uses the last month's (November 2020) data of UNHS 2019/20 as the testing data, and the rest of the previous 10 months' data as the training data. This basic model design covers the breakout of Covid shock in training data and could be helpful to find an outperforming ML model which is less sensitive to the shock in historical data, which gives the researcher more room to use previous data which contains shock. This ex-post model provides guidelines on the choice of features and models as well.

Secondly, to evaluate the model performance and robustness over time, an ex-ante robust design is used. Let the first two months' data be denoted as $Train_0$, and $MonthlyData_i$ is the data of month d,

$$(i, d) \in \{(Index, Dates)\} = \{(1, 2019.11), ..., (4, 2020.02), (5, 2020.07), ..., (9, 2020.11)\}.$$

Then training data and testing data can be defined as:

$$Train_j = Train_{j-1} \cup MonthlyData_j,\ and\ Test_j = MonthlyData_{j+1},\ j \in \{1, 2, ..., 8\}.$$

In other words, the first training data set is the data from the first three months, and the first testing data set is the data from the fourth month, then in each round of new fitting, the next month's data is added to the training data set, and the second month's data becomes the testing data set. This ex-ante design covers the pre- and during Covid data in the testing data, and it simulates a real-world practice of ML in food insecurity prediction when facing a sudden shock.

6.2 ex-post robust design

6.2.1 Evaluation with classification metrics

In the ex-post robust design, the main task is to identify the "Poor" food security households, thus, FCS = 21 is chosen as a threshold of food insecurity, and households with FCS ≤ 21 are in class 1, and the rest of food secure households are in class 0, the class weights during model training are generated inverse proportionally to the number of observations in each class. ML algorithm's performances are evaluated by the AUC of ROC, thus, all cross-validation performances are evaluated by AUC. Figure 7 shows the ROC AUC of four ML algorithms, the XGBoost outperforms other models, and the SVM performs the worst. Table 3 shows the model performances of the default output for the Python sklearn package. To be specific, for logistic regression, random forest, and XGBoost, the model output is (or equivalent to) the probability to be class 1 (food insecure), the default decision threshold is 0.5, i.e. for $p(\hat{y} = 1|x_i) > 0.5,\ \hat{y} = 1$, besides, the output for SVM is not in the form of probability, the default output is the best separation of class 1 and 0.

Models	accuracy	precision	recall	f1	AUC
Logistic	0.748	0.142	0.769	0.240	0.811
Random Forest	0.799	0.150	0.615	0.241	0.819
XGBoost	0.818	0.164	0.615	0.259	0.838
SVM	0.744	0.130	0.692	0.219	0.794

Table 3: Model performance for default output

In practice, depends on the benefit v.s. cost trade-off, one can adjust the decision threshold to a probability different from 0.5, Figure 8 shows the metrics trade-off. The model

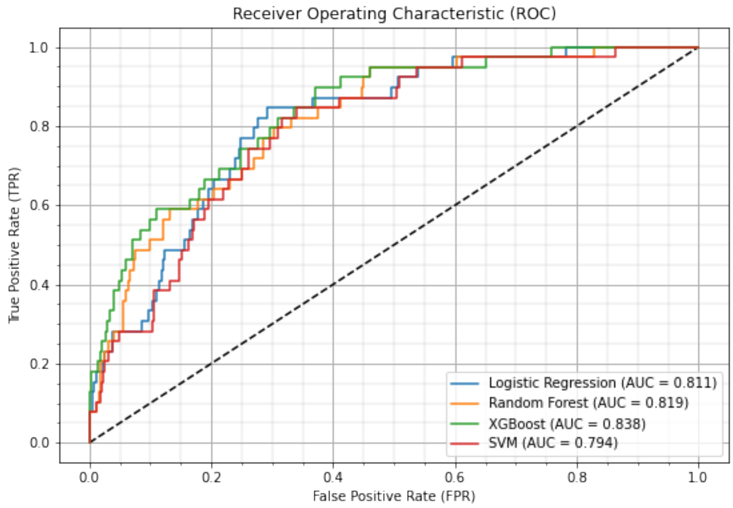


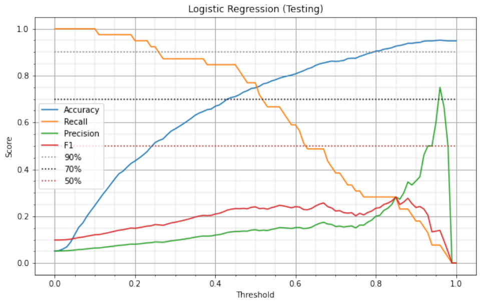
Figure 7: ROC for Testing Data

accuracy and recall curves of SVM (Figure 8.d) change sharply with respect to the threshold probability of less than 0.1, this gives the decision maker a narrow space to adjust the threshold and trade-off between benefit and cost. The algorithm of SVM targets the best separation hyperplane instead of probability, thus, the result of SVM in Table 3 could be regarded as the best performance of SVM. On the contrary, the performance of regularized logistic regression and tree-based methods (Figure 8.a, 8.b and 8.c) gives the decision maker much more flexibility, this shows the power of high AUC ROC. To show the trade-off more specifically, in Figure 8 the dashed horizontal lines are given, for given recall equals 50%, 70%, and 90%, policymakers can read the corresponding accuracy, precision, and f1 score to make their decisions. The high AUC model gives policymakers greater flexibility. In addition, in the interval of relatively high accuracy and recall, the precision score is quite low, this shows the trade-off between type I and type II errors and is caused by the imbalance of data.

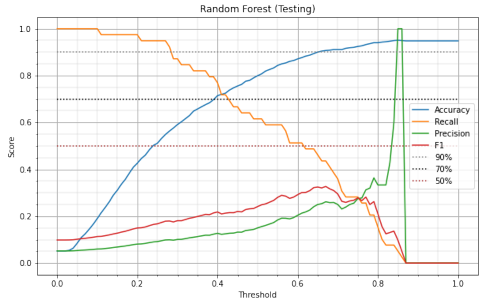
Therefore, we can conclude that tree-based models are more robust against a shock, and the XGBoost outperforms the others. SVM cannot reveal the complex interaction among so many predictors, and regularized logistic regression benefits from the feature selection $l1$ regularization and performs also robustly, but worse than tree-based methods. Meanwhile, ensemble or boosting tree models such as random forest or XGBoost can capture those interactions and shock dynamics.

6.2.2 SHAP interpretation

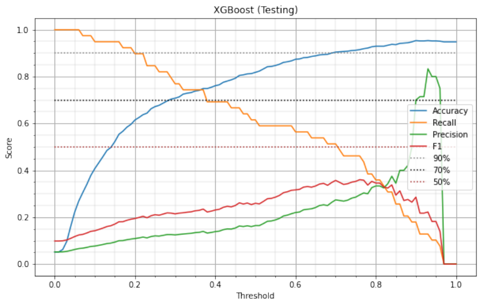
The SHAP method and its Python package enable us to inspect the feature contributions of food insecurity prediction. Meanwhile, for tree-based models, feature importance scores can be computed, it is calculated by the contributions of each predictor when growing



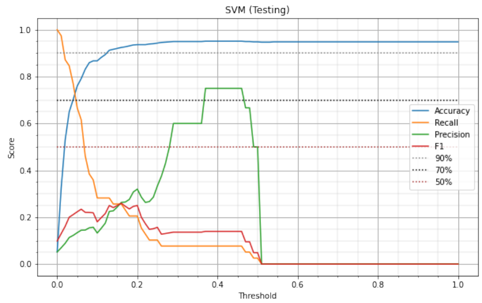
(a) Logistic Regression



(b) Random Forest



(c) XGBoost



(d) Support Vector Machine

Figure 8: Performances with different threshold

decision trees. However, the caveat of SHAP and feature importance score is that they do not imply any causality.

Figure 9 shows the feature importance scores for all predictors in the random forest model, the scores are calculated by the mean decrease of node impurity. Not surprisingly, the household-level data contributes the most, some demographical features such as family size, the types of staple food the household grows, and the male ratio in the household have strong predictive power. Meanwhile, some wealth features also contribute to the prediction, such as the value of total assets and furniture.

However, the feature importance score does not reveal how each feature influences the prediction. SHAP value is helpful in exploring greater details about ML models. For this food insecurity problem, a higher SHAP value for an observation brings a higher probability of being food insecure (class 1). Figure 10 shows the top 20 influential features of our tree-based models. Random forest (10.a) explores the most in the household-level data, and the top features are similar to the result in Figure 9. XGBoost explores more information on the macro-level data, for instance, the precipitation data and temperature data, and it also utilizes the information of previous UNHS 2016/17 FCS district-level distribution. In addition, the SHAP of logistic regression also shows that it utilizes UNHS 2016/17 FCS data (not shown in figures), this may explain why the AUC of logistic regression and random forest are close, because logistic regression may explore this data

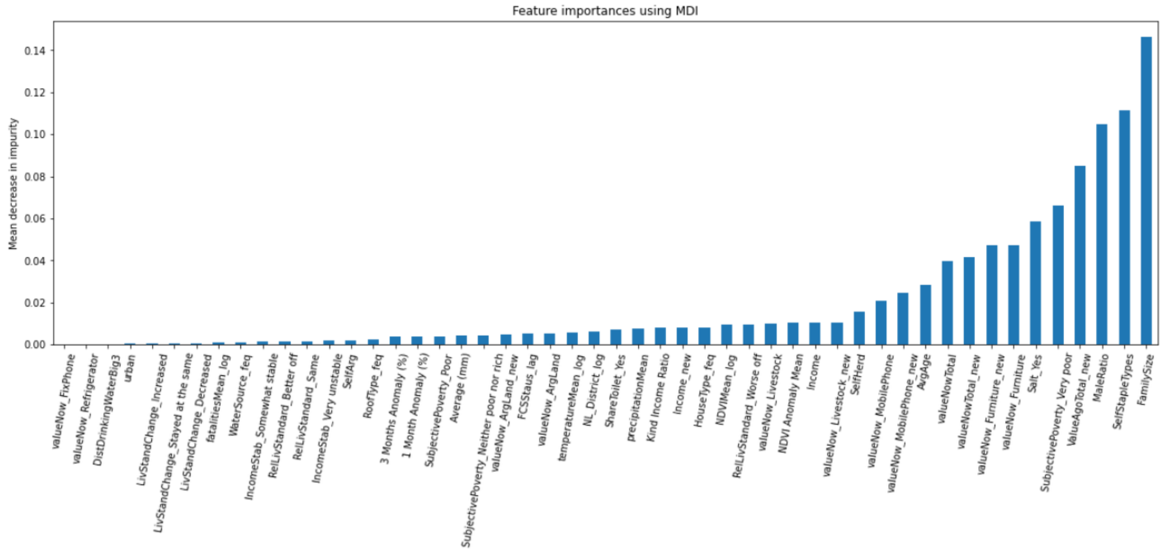


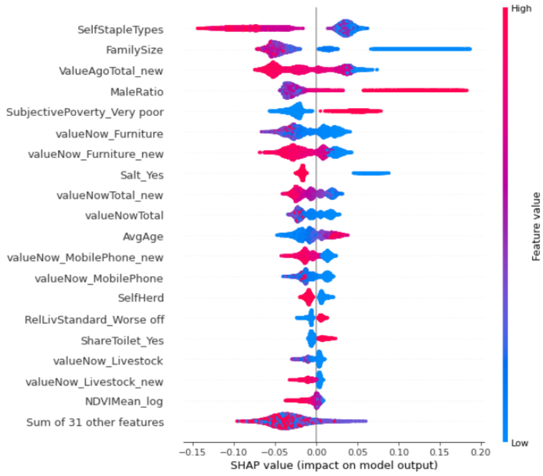
Figure 9: Feature Importance of Random Forest

more.

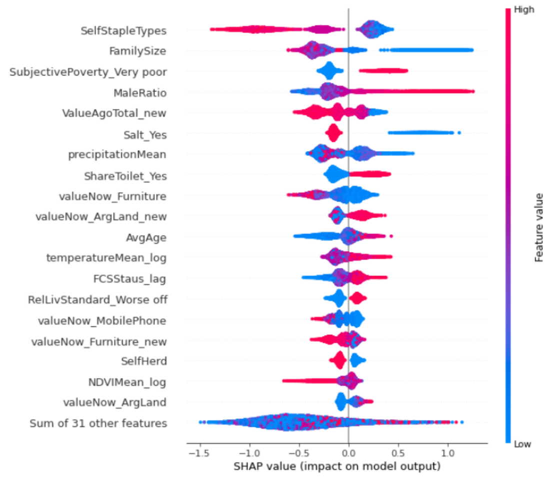
Furthermore, SHAP can be used to understand the impact of each feature, similar to how the coefficients of OLS linear regression are understood. The top two of both tree-based models are the *types of staple food a household grows* and the *family size*, the more types a household grows (higher value indicated by the color), the more possible this household is food secure (negative side of 0 in the horizontal axis), and vice versa. The smaller the family size, the higher the probability of food insecurity. For binary variables such as subjective poverty of very poor, the 1 class (red) increased the probability of being food insecure, and class 0 (blue) vice versa. Figure 10.c and 10.d shows the SHAP values of the testing data, and some of the top features ranking are changed, this may be caused by the difference in the training and testing datasets.

SHAP also enables researchers to understand the interactions of variables, as shown in Figure 11. The horizontal axis is the value of one predictor, and the vertical axis is the corresponding SHAP value, positive SHAP increases the risk of food insecurity, and negative SHAP decreases it. In addition, the color bar shows features interaction. The scatter plot can reveal the non-linear relationship between the predictors and the response variable. For example, the impacts of precipitation and total assets value (one year ago) are non-monotone, and the impacts of average age and types of staple food grown by households are consistent with Figure 10.

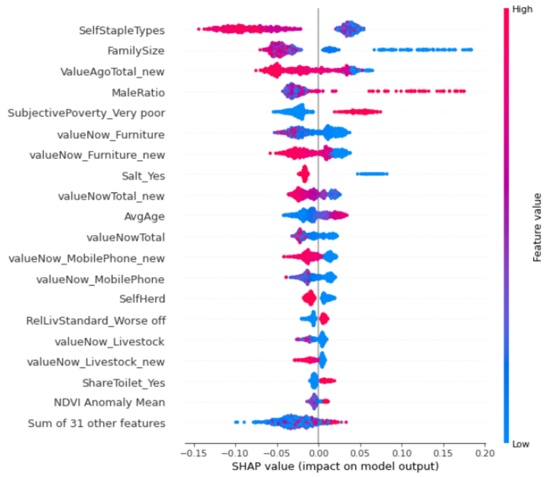
Finally, SHAP is also inspiring in the feature selection step. After training the model, one can use the SHAP plot as Figure 10.a or 10.b, and drop the less influential features. By selecting important features, the noise and dimensionality of data are reduced, and the model performance could be improved. Nevertheless, to prevent data leakage, only



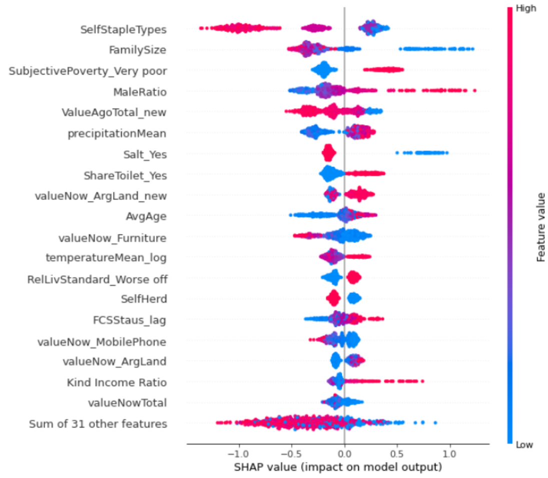
(a) Random Forest (training data)



(b) XGBoost (training data)



(c) Random Forest (testing data)



(d) XGBoost (testing data)

Figure 10: Top 20 SHAP Value for tree-based models

the SHAP information of the training data is used here.

6.2.3 Identifying borderline food insecurity

In addition to identifying the "Poor" food security households, this subsection tries to use the same model and features to identify the 'Borderline" food security households, i.e. FCS = 35 is the cut-off for food insecurity. The results are shown in the appendix. Compared with the previous results, the AUC ROC decreases, and the tree-based methods still outperform. Surprisingly, random forest performs similarly to logistic regression. Due to the reduction of data imbalance, the precision scores are improved, which means the type I error decreases. However, the accuracy performs worse than in the previous case. Features are selected according to the distribution and model training performance of FCS = 21 as the threshold, a more elaborate feature engineering and tuning might improve the model performance, but this scenario will not be focused further in this study.

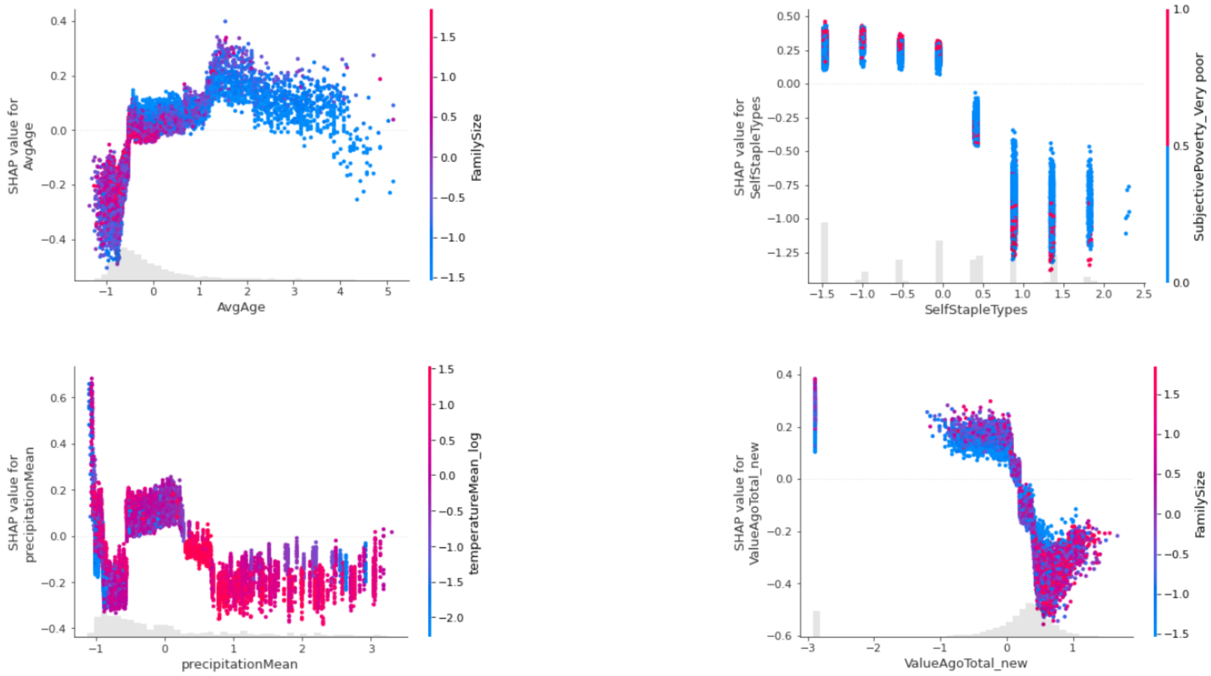


Figure 11: Scatter Plot of variable interactions

6.2.4 Resampling methods

As mentioned in Section 2, resampling methods such as SMOTE and ADASYN could also overcome the data imbalance. In order to mitigate the curse of dimensionality, only 10 important features are selected. The selection process considered the SHAP value of Random Forest and XGBoost, and in Figure 12 there is no strong correlation within 10 features. Notably, although resampling with fewer features might improve the sensitivity of SVM, the inflexibility and the above poor performance of SVM leads it to be dropped. Figure 13 shows that the AUCs of both tree-based models decreased, and the performance of logistic regression improved. The performance deterioration of tree-based methods shows that the resampling methods perform worse than the weighted class scheme. For the weighted class scheme as above, class weights that are inversely proportional to the number of observations are considered when tree splits are made. On the contrary, resampling methods make the data become balanced, which only gives tree models equal weight within the minority class and majority class. The improvement of logistic regression is caused by the nature of logistic regression, it fits the model with respect to the whole distribution, and when the data becomes more balanced, the food insecure class is more distinguishable. In conclusion, resampling methods for tree-based models do not help to identify the shock in training data compared with the weighted class scheme. However, Resampling helps logistic regression to learn more about the whole training data and performs more robustly.

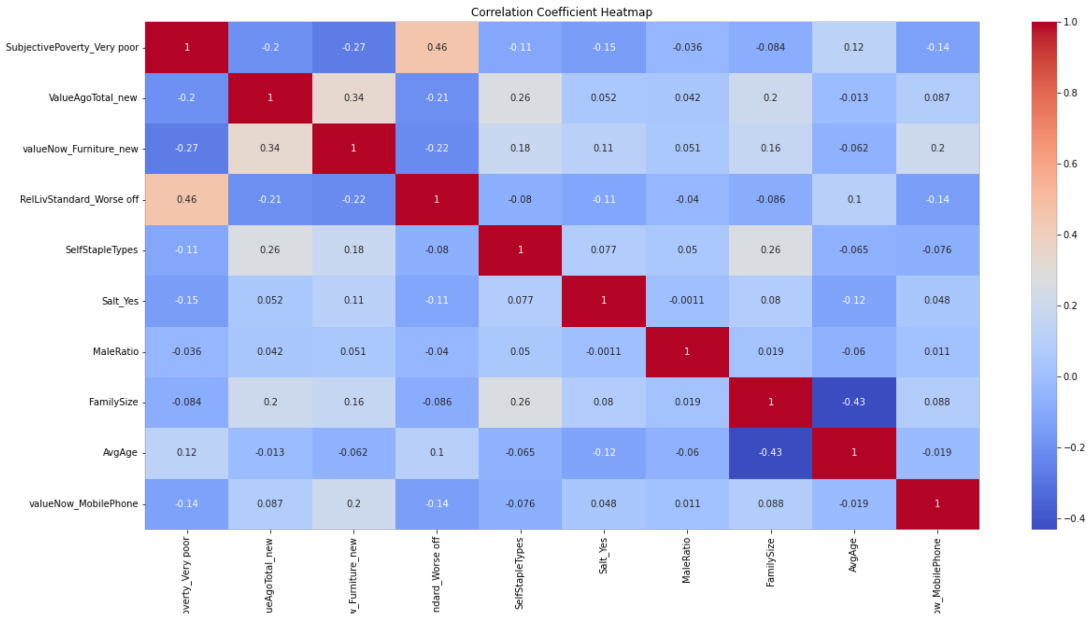


Figure 12: Correlation Coefficients of top 10 features

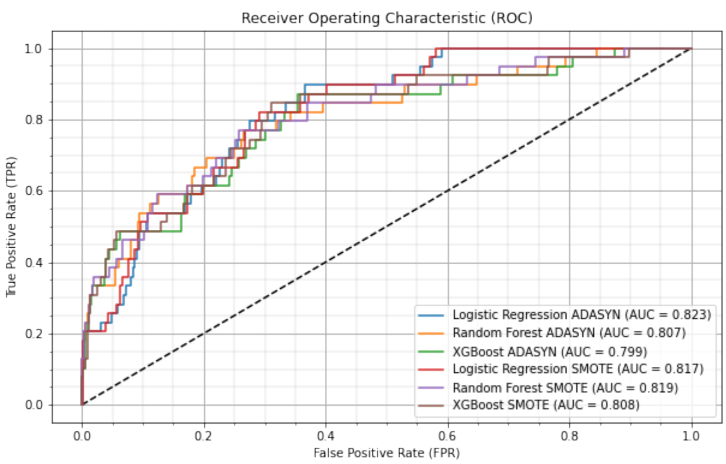


Figure 13: ROC and AUC for two resampling techniques

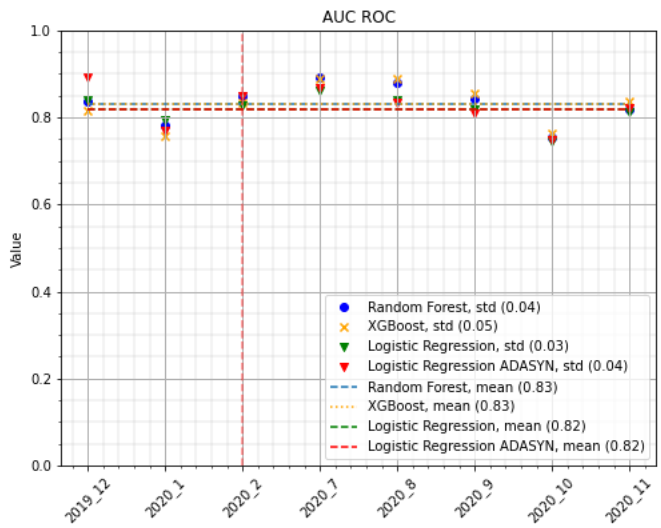


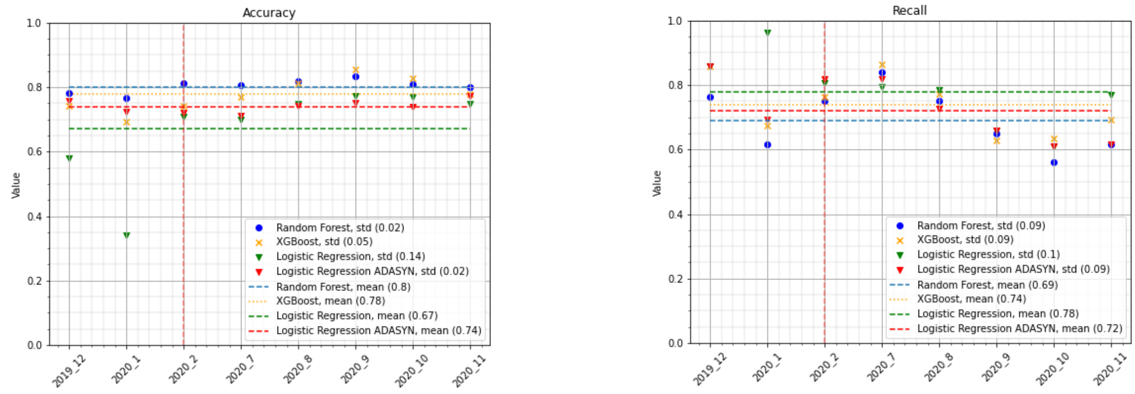
Figure 14: AUC

6.3 ex-ante robust design

6.3.1 Performance evaluation

As in the ex-post design, the target households are also those FCS ≤ 21. From the ex-post result, we find that the tree-based models and logistic regression with resampling have relatively high AUC. Thus, random forest, XGBoost, and logistic regression (with and without ADASYN resampling) are chosen in this section. Data before March 2020 is regarded as data pre-Covid, and data after that is regarded as during Covid. Then we can compare the model performances when predicting monthly pre- and during Covid.

Figure 14 shows the AUC of each model and the overtime AUC mean and standard deviation. Firstly, the ACU for the four models in most months are similar. XGBoost gives a relatively higher AUC compared with others during the Covid, but performs worse than others before the Covid breakout, this implies that XGBoost requires a bigger data volume. The second best performed is the random forest, and it performs similarly to XGBoost during the breakout of Covid. When focusing on the performance of July and August 2020, which is the month of the breakout of Covid in UNHS data, the two tree-based models slightly outperform the logistic models. This shows the flexibility and robustness of tree-based models when new patterns (shock) come. On the contrary, logistic regression models rely on the overall distribution of our data, and in high dimensional cases, it is less robust against a shock. The deterioration of logistic regression's performance ranking before and after July 2020 shows this. Therefore, we can conclude that the tree-based methods could mitigate the shock in the testing data. Meanwhile, the more data tree-based models are fed, the better the performance. On the contrary, logistic regression models are less flexible, and when facing a shock, they cannot distinguish and identify the shock, although they perform similarly to or even better than tree-based



(a) Accuracy (b) Recall

Figure 15: Default output of accuracy and recall

models in pre-Covid months.

In addition, Figure 15 shows the default model output ($p(\hat{y} = 1|x_i) > 0.5 \Rightarrow \hat{y} = 1$) of accuracy and recall for these models, the data is in the appendix. The accuracy performances of tree-based models are much more robust and good. Although the recall performances of logistic models are over the tree-based models, a slightly higher AUC of tree-based models enables them to be adjusted.

6.3.2 Bootstrap robustness

The above only uses available real data, but it cannot provide statistical distributional information on metrics. This limits the further inspection of model performances and their robustness. Nevertheless, the famous Bootstrap method could provide some more information on this. Bootstrap in this section samples the monthly testing data set with replacement B times, $B = 500$, and the trained model predicts this generated new testing data B times. Then B metrics are obtained, and the distribution of these metrics is a proxy of the true distribution of the testing metrics. Moreover, by comparing the standard deviations of metrics, the robustness of models could also be partially revealed.

Figure 16 shows the bootstrapped 95% confidence interval of AUC, the points of monthly AUC are estimated by original testing data. The confidence interval is the pivotal interval (more to see Wasserman (2004)), this type of confidence interval can ensure the confidence interval always covers the original parameter $\hat{\theta}$. It is $(2\hat{\theta} - \hat{\theta}^*_{1-\alpha/2},\ 2\hat{\theta} - \hat{\theta}^*_{\alpha/2})$, here $\hat{\theta}$ is the estimated metric from original testing data, and $\hat{\theta}^*_a$ is the ath percentile of the B bootstrapped estimated metrics. From Figure 16, the AUC of tree-based models and of logistic models are not significantly different in July 2020, but it shows a greater difference between them in August and September 2020. In October and November 2020, the differences are not significant anymore. Tree-based models perform better facing the shock, but logistic regression with resampling even significantly outperforms against XG-

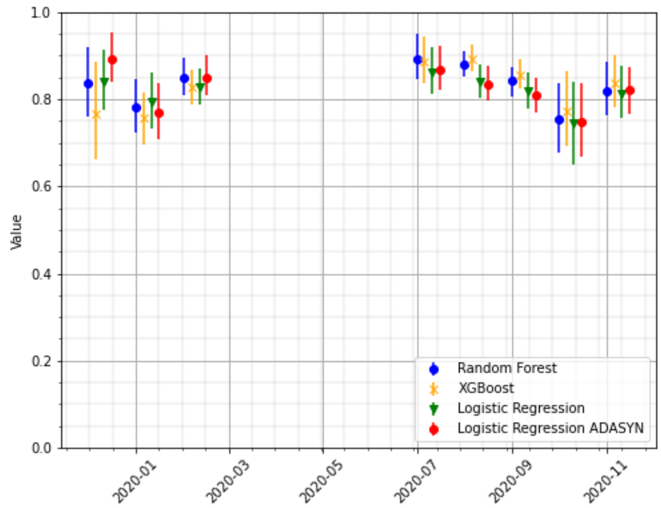


Figure 16: AUC with bootstrap

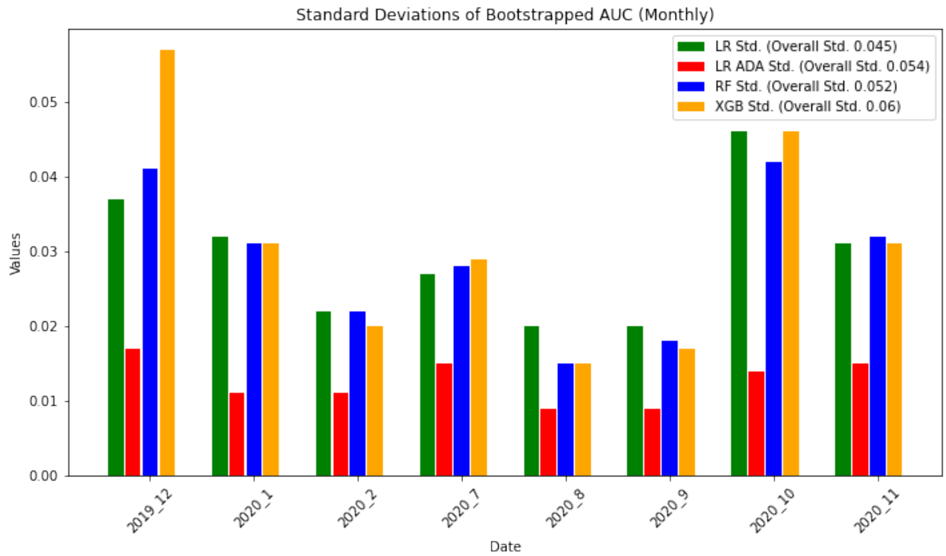
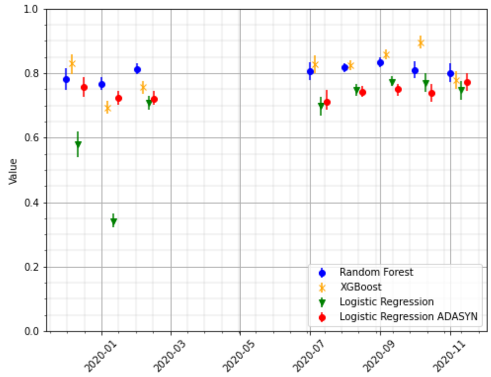


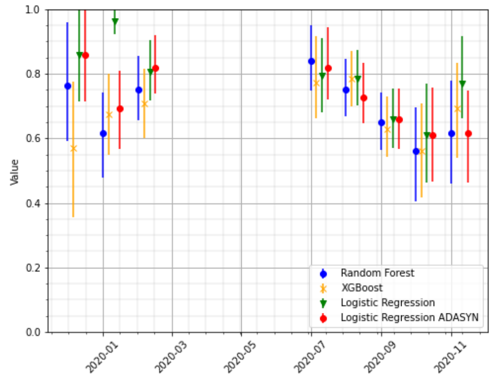
Figure 17: Standard deviation of bootstrapped AUC

Boost in December 2019. Meanwhile, before Covid, with a limited volume of training data, the tree-based models perform worse, especially for XGBoost in December 2019. This result is consistent with the result of the original testing data above.

In addition to the pivotal confidence intervals, Figure 17 shows the standard deviation of bootstrapped AUC for each month, and the overall standard deviation on the label is computed by all bootstrap samples for all months. Although the overall standard deviation of logistic regression with the ADASYN sample is higher than without ADASYN, the standard deviations within each month of ADASYN logistic are surprisingly the lowest, this also confirms the great improvement of dimension reduction techniques on logistic regression. However, this improvement is not simply caused by the dimension reduction, because when compared with the AUC standard deviation of the ADASYN random forest shown in the appendix, the feature selection and resampling do not reduce the standard



(a) Accuracy



(b) Recall

Figure 18: Default output of accuracy and recall

deviation of the random forest. Furthermore, the XGBoost performs similarly with random forest and logistic regression in terms of standard deviation except in December 2019.

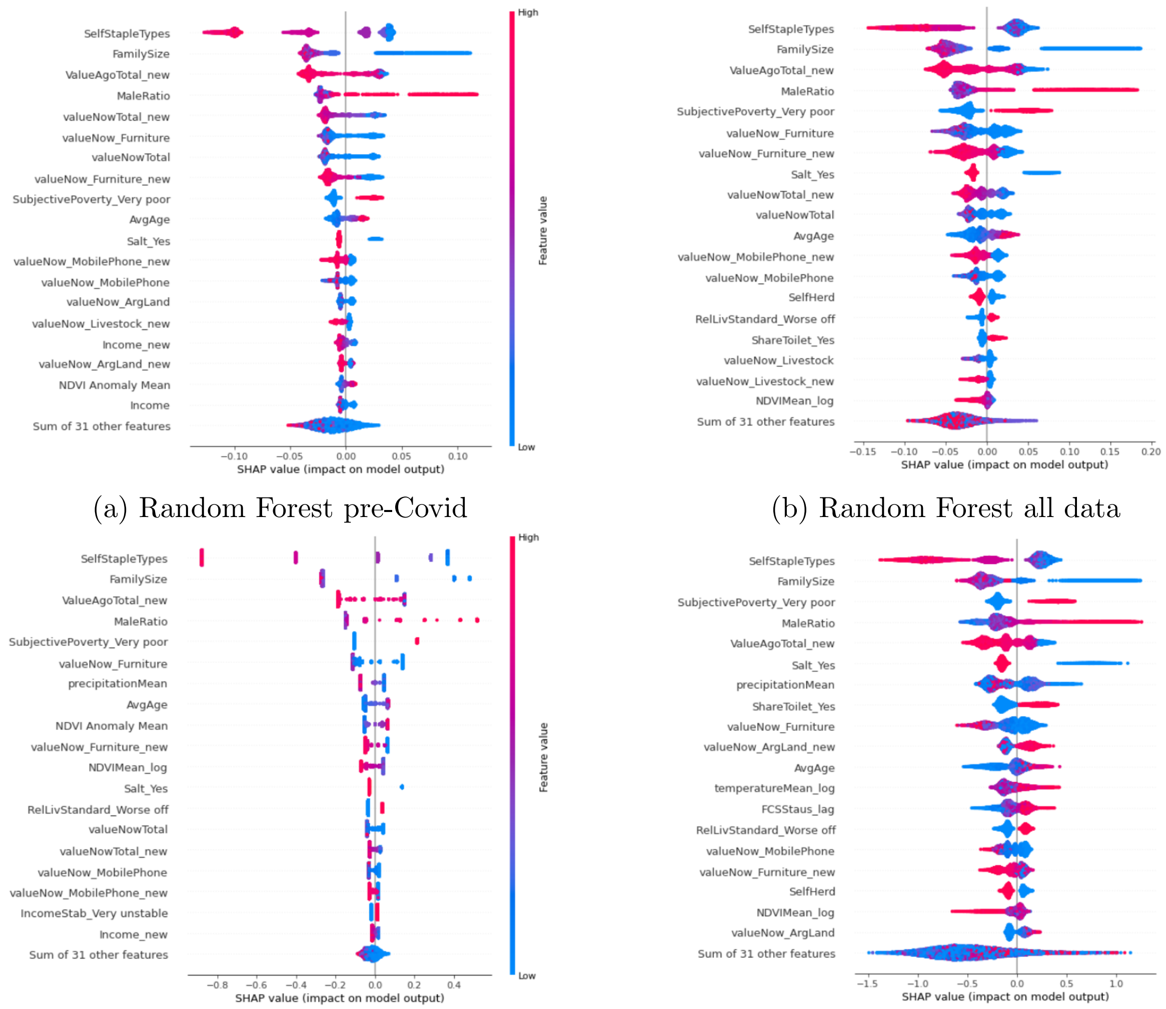
Figure 18 shows the default output accuracy and recall with 95% confidence interval. As shown in the ex-post model, tree-based methods do not perform significantly better in terms of recall with the default output. Nevertheless, the accuracy of tree-based models in the early breakout of Covid (July to September 2020) is significantly higher than logistic models.

6.3.3 SHAP analysis

Is it possible that the shock of Covid changes the feature importance of food insecurity prediction? Figure 19 shows the top 20 features given by SHAP value for models trained with and without during Covid data. The best predictor is *the number of types of staple food a household grows*, demographic features and asset features also play a critical role in prediction. When focusing on the random forest (Figure 19.a and 19.b), the importance ranking for some asset indicators changed, and during Covid, *if one household has salt* and *if one household shares toilet with others* becomes more important predictors for food security prediction, this also happens in the XGBoost model. When focusing on the XGBoost (Figure 19.c and 19.d), more macro-level features play a big role compared with random forest. The SHAP feature importance ranking changed after the shock of Covid, but there are still some features that work robustly, for example, *types of crops grown by a household*, *male ratio*, and *family size*.

6.3.4 Predict with the same training data

Because the sampling method of UNHS is not i.i.d. in the time dimension, the stochastic volatility and time series trend are difficult to be distinguished in UNHS data. Thus, to



(a) Random Forest pre-Covid

(b) Random Forest all data

(c) XGBoost pre-Covid

(d) XGBoost all data

Figure 19: SHAP for pre- and during Covid

show the robustness of ML models, it makes sense to use the same set of training data to train and test models. All data for pre-Covid is used to train ML models and the performances of during Covid period is shown in Figure 20 and 21.

The behaviors of ML models when only trained by pre-Covid data are similar to the updated training data case. In terms of AUC, tree-based models still outperform in July, August, and September, which shows their outstanding robustness and flexibility when facing a shock in new data.

6.4 Low-cost ex-ante model

Data availability is an important factor needed to be considered in the food insecurity issue because micro-level data collection requires much more effort and cost for governments and organizations. For example, Lentz et al. (2019) compared the performances of linear regression given different scales and availabilities of data. This section will make the food insecurity prediction more practical, only a few household-level micro features

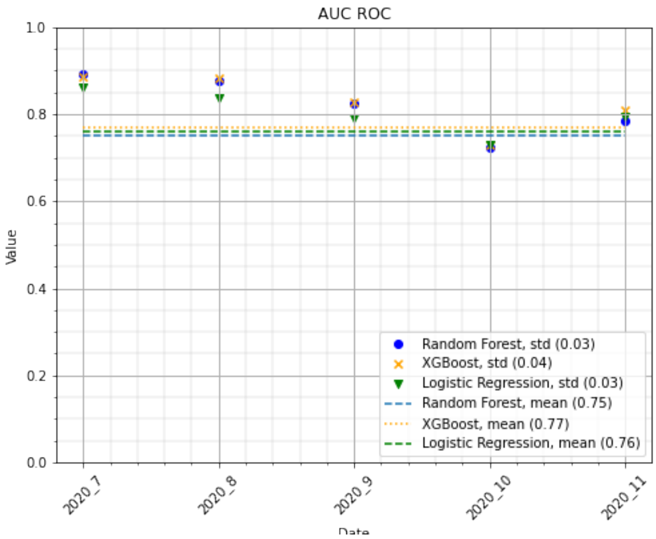
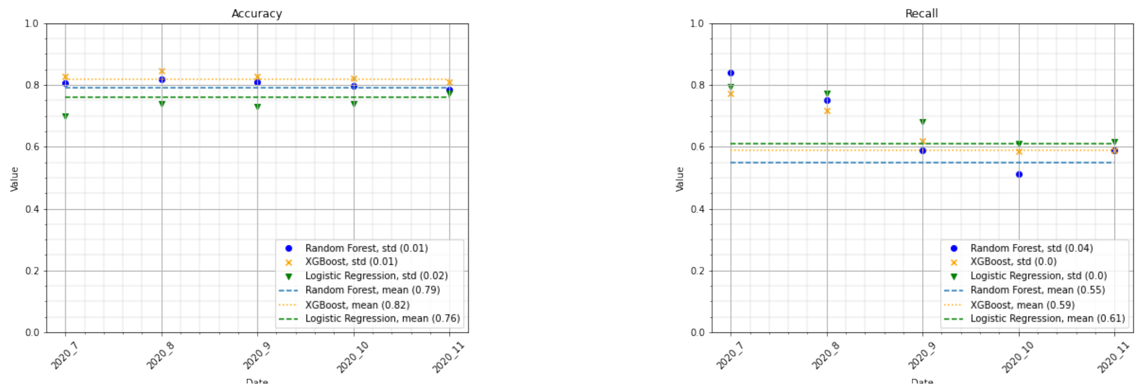


Figure 20: AUC for during Covid

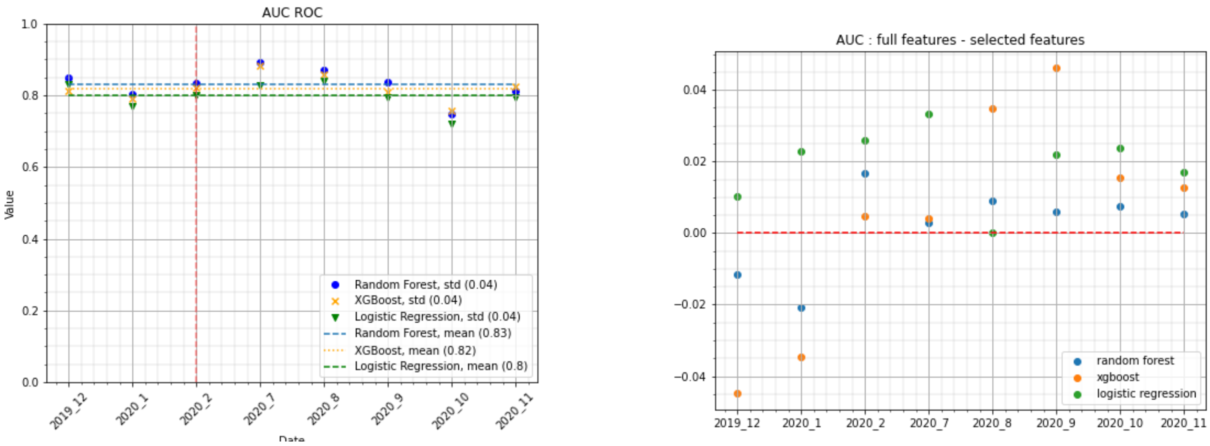


(a) accuracy (b) recall

Figure 21: Default accuracy and recall during Covid

will be used as predictors, and all open-source data will be used. Then governments and institutions could conduct surveys and food insecurity predictions economically. The micro-features that appear in this section are selected according to the previous SHAP sections.

The selected micro features are: *family size*, *the average age of household*, *how many males in the household*, *if the household owns or grows livestock*, *if the household has salt now*, *how many types of staple food crops the household grows*, *subjective poverty*, *relative living standard compares with neighbors*, *living standard change in the last 12 months* and *if one household shares toilet with others*. Only above ten household features needed to be collected (here *urban* is regarded as macro data), and combined with macro open-source data, the ML models could be trained for prediction. Unlike asset indicators, these variables can be collected much easier in a survey.

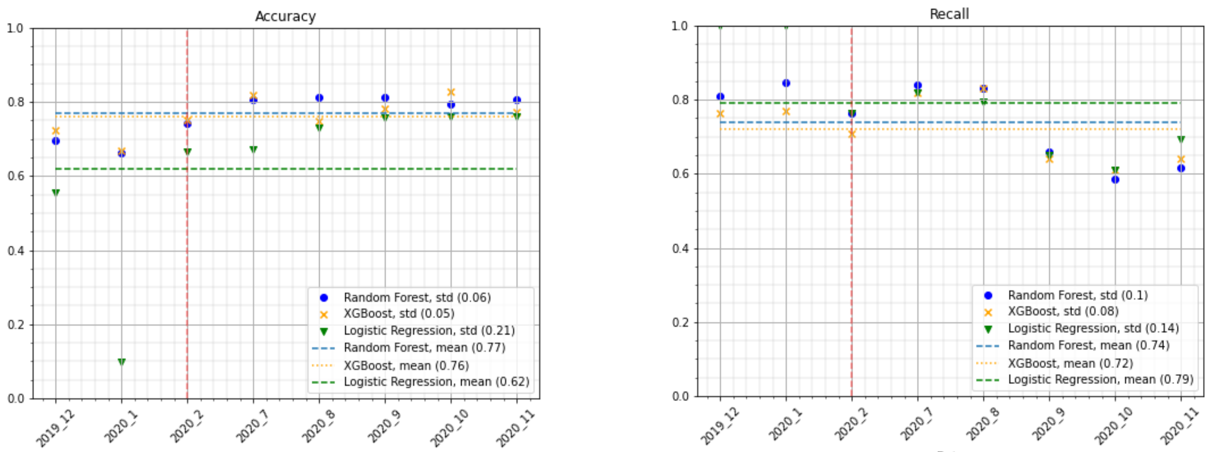


(a) AUC with selected features

(b) Difference of AUC

Figure 22: Performances with selected features

Figure 22.a shows the performances of ML models with selected features. Focusing on July 2020 and later, the AUC performances and trends are similar to the previous full features section. The overall performance of tree-based methods is better than logistic regression, even in the pre-Covid periods. At the beginning of the Covid breakout, tree-based methods achieve 0.8 to 0.9 AUC. Figure 22.b shows the difference between the full feature version and the selected feature version, a positive value means the full feature version AUC is bigger than the selected version AUC. One can find that the deviations of random forest is small and stable during Covid, but deviations of XGBoost are high in August and September 2020. Before Covid, tree models even achieved higher AUC with fewer features. Considering the absolute performance, one can conclude that the selected features also guarantee the performances of ML models, and random forest performs more robustly.



(a) accuracy

(b) recall

Figure 23: accuracy and recall with selected features

Figure 23 shows the default output accuracy and recall of the selected feature version models, the trends are similar to the full feature versions', and the recall performance of tree-based methods even better in August. In addition, if remove all open-source data and only use the selected ten features, ML models also perform similarly on AUC, the result could be found in the appendix, and the robustness conclusion still holds. However, this does not undermine the importance of macro open-source data, the SHAP importance value of XGBoost in the appendix shows open data still helps (precipitation).

6.5 Summary and discussion

In summary, tree-based algorithms are more robust when facing a shock, because they can reflect the complex interactions in their models, which is less sensitive to a shock. On the contrary, logistic regression and SVM have to build models by the overall distribution, thus being more sensitive to outliers in data. Logistic regression performs similarly to tree-based models, but after the shock tree-based models outperform. Meanwhile, household-level micro features provide the most prediction power in our models, the nature of our household-level prediction determines this, but macro-level data also make contributions. Finally, either using the same pre-shock training data to predict during shock food insecurity or using much fewer features to predict, tree-based models always show advantages in robustness, and random forest outperforms.

Furthermore, in the previous ML food insecurity studies, there are two most comparable papers for this study. Zhou et al. (2022) used gradient boosting and random forest with resampling methods to predict the FCS in Uganda, and they achieved AUC of less than 0.65 for all methods, which is less than the mean AUC of 0.83 in ex-ante tree-based models. Meanwhile, when fixing recall to 70%, Zhou et al. (2022) only achieved less than 60% accuracy in Uganda, which is less than the July and August 2020 results of ex-ante models. However, for other African countries, their metrics performed better, for example, 0.83 of AUC in the Tanzania FCS case. In addition, in their study, resampling methods improved the performance of random forest, but our result does not show this, perhaps this difference is caused by technical differences in model design. Regarding the key features, Zhou et al. (2022) also found that assets and geographic indicators are helpful for FCS predicting in Uganda, in addition, price and weather features are also good predictors in some African countries. This is consistent with the SHAP results in this study, although price information is not included. Last but not least, Zhou et al. (2022) mentioned that they did not consider the sudden shock in their study, and our study could reduce this gap to some extent.

Meanwhile, Lentz et al. (2019) also used FCS as one of the food insecurity indicators, and they studied the case in Malawi. They have achieved a recall of 0.83 and an accuracy of 0.76, which is still good compared with the default output in this study. Lentz et al. (2019) showed that the assets and demographic data improve the prediction a lot, which inspired the feature selection in this study as well. Notably, Lentz et al. (2019) only used linear regression to predict, which showed the power of elaborately selected features.

Besides the above two studies, Okori and Obua (2011) also focused on Uganda, but the data they analyzed was too old, and their decision tree model is suspected to be overfitting. The highest recall they achieved is 0.81, although they had an AUC of nearly 1. In addition, regarding the key predictors, Meerza, Meerza and Ahamed (2021) and Gao et al. (2020) both found that household size, farm-related measures, and financial indicators are important, which is consistent with our SHAP analysis as well.

Finally, further improvements could be made to this study, for instance, by adding price features, conducting a regression study, using more time-series ML techniques, or using various food insecurity indicators. Furthermore, this study shows the trade-off between recall and accuracy by AUC ROC, further study on the f1 score or the precision-recall curve could show this trade-off from another perspective, and possibly obtain different conclusions. Nevertheless, a different dataset may also improve ML model performances and reduces the stress of such trade-off. A further study for comparing the performances of ML and real-world aiding activities would also be inspiring.

7 Conclusion and Outlook

Many countries around the world are facing the threat of food insecurity, and the breakouts of the Covid pandemic and war in Ukraine escalate the already immense food security challenges the world is facing. This underlines how important it is to adequately forecast food insecurity. ML can be key to improving forecasting substantially. Therefore, the robustness of ML in food insecurity prediction must be considered. This study explores the households and open-sourced data to find robust ML models. Our study could be extended to other eastern African countries without much cost.

With the ex-post model design, an AUC of 0.81 to 0.84 could be achieved, a higher AUC implies a broader space for policymakers to trade off the cost and benefit of correctly detecting real food-insecure households. Without elaborated threshold selection, 62% of food insecure households could be detected, and overall 80% households are correctly classified. Furthermore, tree-based models outperform, because tree-based models could reveal the high dimensional interactions among features, and this covers the shock in training data. Meanwhile, resampling techniques do not help much for tree-based models, but logistic regression is improved. This implies that the balanced weight scheme is good enough for tree-based models compared with resampling because the balanced weight scheme emphasizes more on minority class.

In the ex-ante model design, an overtime mean of 0.83 AUC can be achieved, and at the first and second months of the Covid breakout in our data, without further adjustments, 80% of food insecure households and overall households are correctly classified. Meanwhile, it is shown that tree-based methods are also robust when using the same pre-shock training data to predict following during-shock data. Furthermore, when only collecting ten demographic features, the model performances do not deteriorate significantly, and random forest performs the most similarly to the full feature models. These imply that given limited available data, random forest is the most robust one. Moreover, the tree-based models outperform and show good robustness during Covid than the logistic models, especially at the beginning of the shock. Again this demonstrates the complicated essence of food security issues, those high-dimensional interactions have to be considered in future studies.

In addition, SHAP analysis shows that demographic and asset features are important in food insecurity prediction, which confirms that food insecurity is strongly related to poverty and development. The contributions of micro household data keep consistent with other studies. In a study of regional food insecurity, those macro open-sourced data will contribute more. SHAP analysis also gives us policy implications, that one can design a short questionnaire with the help of SHAP analysis, to collect data and predict food insecurity economically and effectively.

In conclusion, this study shows that ML, especially tree-based methods are suitable for food insecurity prediction, because they could reflect the complex interactions of features, and could better reflect the heterogeneity in data. Tree-based methods are robust to extreme values, this guarantees them to some extent robustness when facing a shock such as Covid in future data, because shocks may bring more extreme values. Thus, compared with the famous logistic regression, tree models such as random forest and XGBoost are recommended for future research, and for a limited data availability case, random forest is more robust.

Further studies with different model designs and predictors or an analytical framework for shock robustness ML analysis could be the next step. For policymakers and international organizations, on the one hand, ML classification metrics give them more space to trade off the cost and benefit for food insecure aiding, which shows the great potentiality of ML implementation on food insecurity. On the other hand, interpretable ML analysis such as SHAP enables them to select predictors more efficiently and also provides new insight in confirming the existing food insecurity theories. Finally, our study also shows an alternative approach on crisis study and gives researchers a data-driven option by ML.

Appendix

Figures

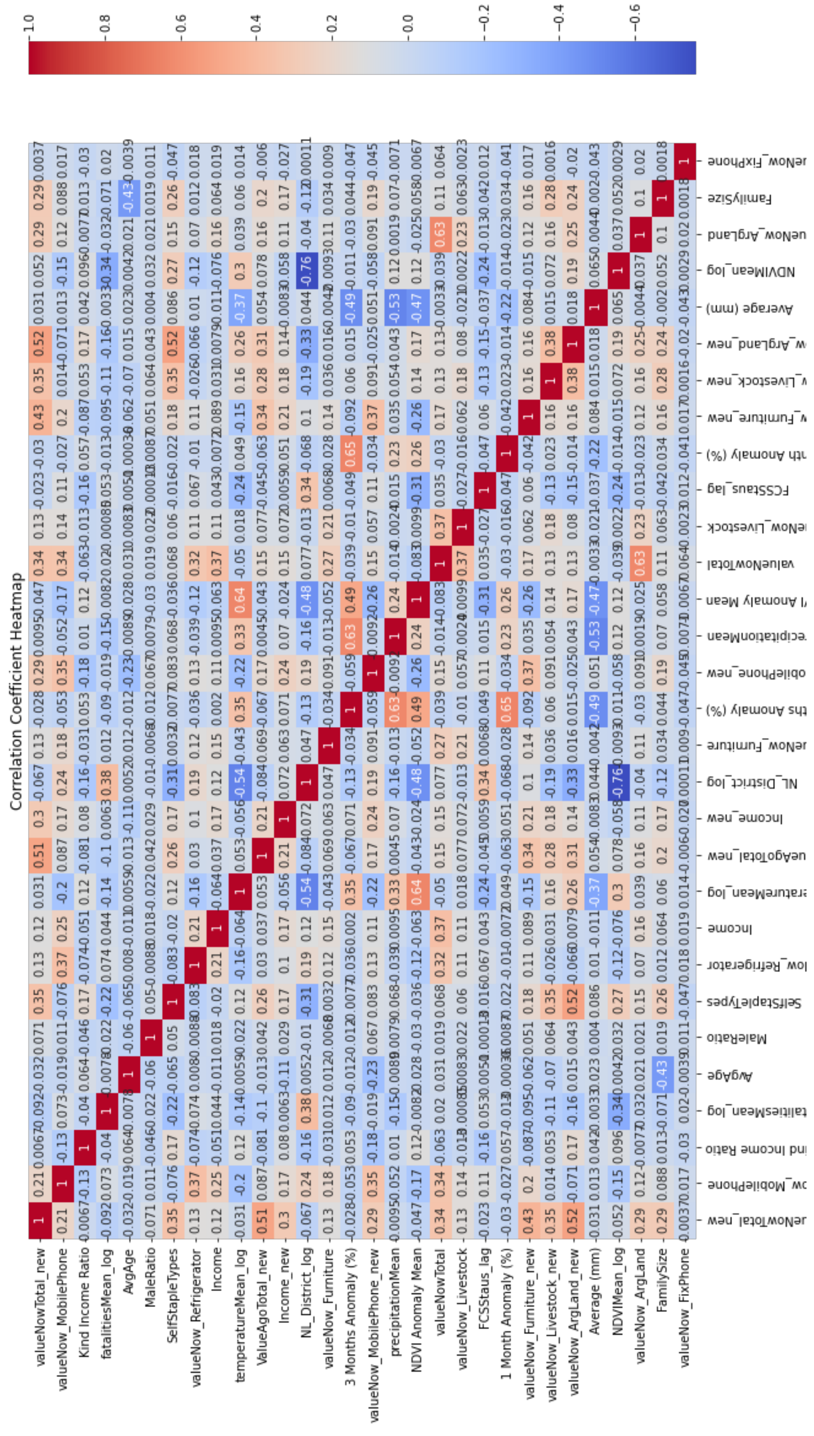


Figure 24: Correlation Coefficient of all continuous features

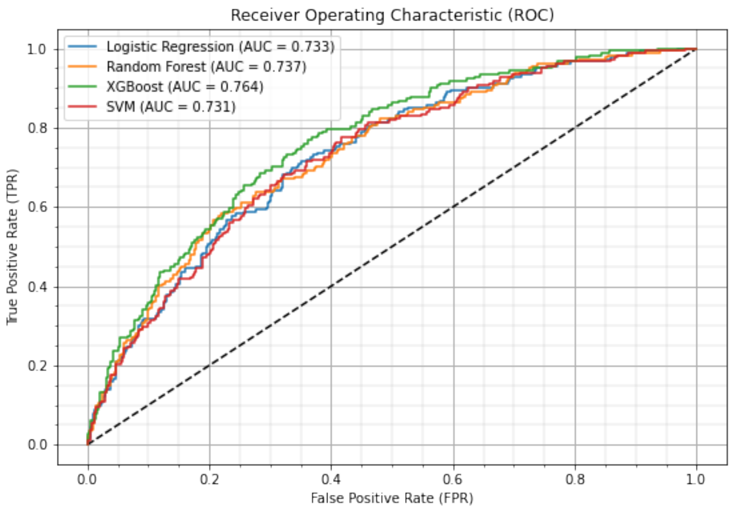
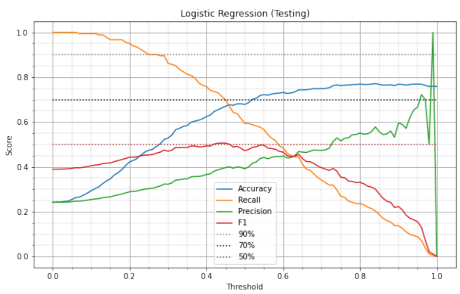
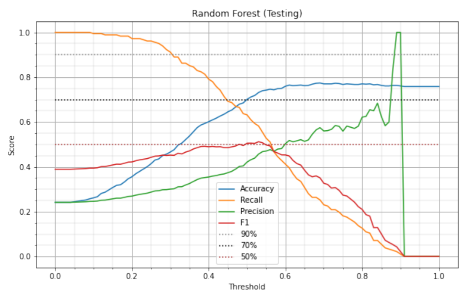


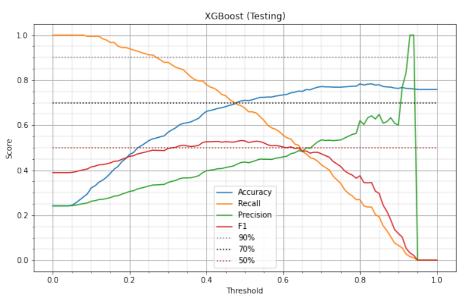
Figure 25: ROC for Testing Data (FCS = 35 as threshold)



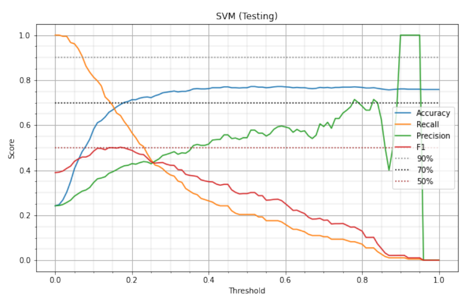
(a) Logistic Regression



(b) Random Forest



(c) XGBoost



(d) Support Vector Machine

Figure 26: Performances with different threshold (FCS = 35 as threshold)

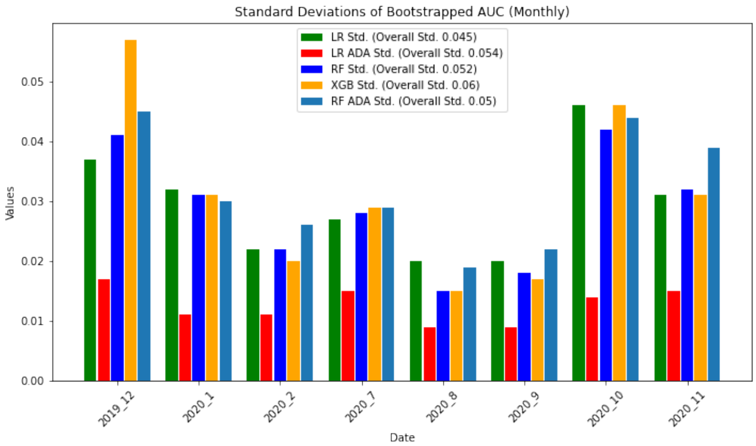


Figure 27: Standard deviation of bootstrapped AUC with ADASYN RF

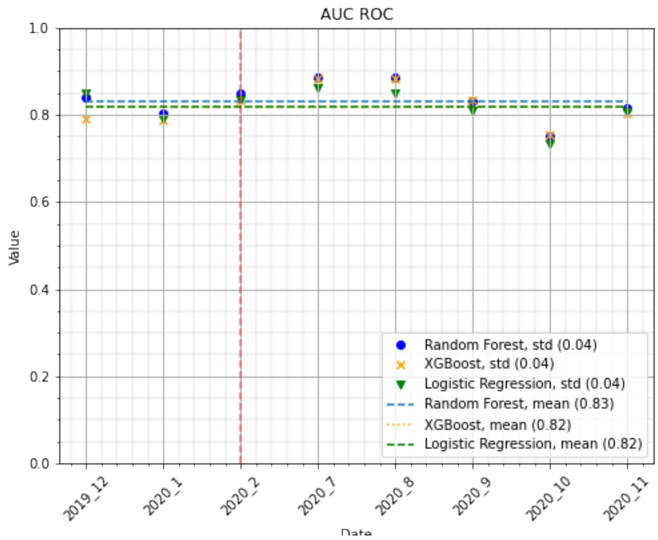


Figure 28: AUC selected features without open-source data

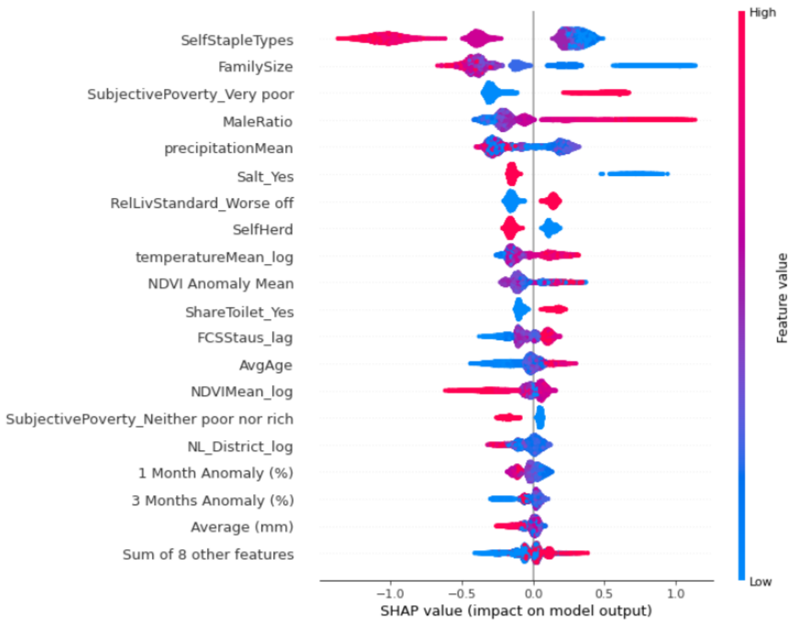


Figure 29: SHAP of XGBoost with selected features and without macro-data

Tables

Variable Name	Type	Treatment
Food insecurity ratio of 2016/17	cont	Std
Total income	cont	Std & log
Income kind given ratio	cont	Std
HH male ratio	cont	Std
HH average age	cont	Std
Type of dwelling	cat	freq
Source of drinking water	cat	freq
Distance of drinking water	cat	freq
Sharing toilet	bin	nan
Having salt	bin	nan
Domestic agricultural	bin	nan
Domestic herding	bin	nan
Income stability	cat	one-hot
Relative living standard	cat	one-hot
Change of living standard	cat	one-hot
Subjective poverty	cat	one-hot
Distance to drinking water	cat	one-hot
Family size	disc	Std
Types of staple food HH grows	disc	Std
Value of mobilephone	cont	Std & log
Value of furniture	cont	Std & log
Value of agricultural land	cont	Std & log
Value of likestocks	cont	Std & log
Total assets	cont	Std & log
Total assets last year	cont	Std & log
Value of refrigerator	cont	Std
Value of fixphone	cont	Std

Table 4: UNHS Predictors

Abbreviations: *Std*: standardization; *freq*: frequency encoding; *one-hot*: one-hot encoding; *log*: log-transformation; *cont*: continuous; *cat*: category; *bin*: binary; *disc*: discrete; *HH*: household

AUC	RF	XGB	LR	LR ADASYN
2019.12	0.837	0.815	0.841	0.892
2020.01	0.782	0.758	0.794	0.769
2020.02	0.850	0.834	0.827	0.848
2020.07	0.894	0.889	0.861	0.868
2020.08	0.880	0.890	0.841	0.834
2020.09	0.842	0.855	0.817	0.808
2020.1	0.755	0.764	0.745	0.749
2020.11	0.819	0.838	0.811	0.823
recall	RF	XGB	LR	LR ADASYN
2019.12	0.762	0.857	0.857	0.857
2020.01	0.615	0.673	0.962	0.692
2020.02	0.750	0.764	0.806	0.819
2020.07	0.841	0.864	0.795	0.818
2020.08	0.750	0.773	0.784	0.727
2020.09	0.650	0.630	0.660	0.660
2020.1	0.561	0.634	0.610	0.610
2020.11	0.615	0.692	0.769	0.615
accuracy	RF	XGB	LR	LR ADASYN
2019.12	0.782	0.742	0.580	0.758
2020.01	0.768	0.694	0.342	0.725
2020.02	0.814	0.741	0.708	0.722
2020.07	0.807	0.769	0.699	0.711
2020.08	0.817	0.812	0.748	0.743
2020.09	0.834	0.855	0.774	0.750
2020.1	0.810	0.829	0.769	0.739
2020.11	0.799	0.780	0.748	0.773

Table 5: ex-ante performances and default output

References

Andree, Bo Pieter Johannes. 2022. "Machine Learning Guided Outlook of Global Food Insecurity Consistent with Macroeconomic Forecasts.".

Andree, Bo Pieter Johannes, Andres Chamorro, Aart Kraay, Phoebe Spencer and Dieter Wang. 2020. "Predicting food crises.".

Barbosa, Rommel Melgaço and Donald R Nelson. 2016. "The use of support vector machine to analyze food security in a region of Brazil." *Applied Artificial Intelligence* 30(4):318–330.

Browne, Chris, David S Matteson, Linden McBride, Leiqiu Hu, Yanyan Liu, Ying Sun, Jiaming Wen and Christopher B Barrett. 2021. "Multivariate random forest prediction of poverty and malnutrition prevalence." *PloS one* 16(9):e0255519.

Chen, Tianqi and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. pp. 785–794.

Cooper, Matthew, Benjamin Müller, Carlo Cafiero, Juan Carlos Laso Bayas, Jesús Crespo Cuaresma and Homi Kharas. 2021. "Monitoring and projecting global hunger: Are we on track?" *Global Food Security* 30:100568.

Department of Economic and Social Affairs. 2022. *Progress towards the Sustainable Development Goals 2022*. United Nations.
URL: *https://sdgs.un.org/goals/goal2*

Elvidge, Christopher D, Kimberly E Baugh, Mikhail Zhizhin and Feng-Chi Hsu. 2013. "Why VIIRS data are superior to DMSP for mapping nighttime lights." *Proceedings of the Asia-Pacific Advanced Network* 35(0):62.

Fernández, Alberto, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk and Francisco Herrera. 2018. *Learning from imbalanced data sets*. Vol. 10 Springer.

Gao, Chen, Chengcheng J Fei, Bruce A McCarl and David J Leatham. 2020. "Identifying Vulnerable households using machine learning." *Sustainability* 12(15):6002.

Hastie, Trevor, Robert Tibshirani, Jerome H Friedman and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2 Springer.

Hossain, Marup, Conner Mullally and M Niaz Asadullah. 2019. "Alternatives to calorie-based indicators of food security: An application of machine learning methods." *Food policy* 84:77–91.

James, Gareth, Daniela Witten, Trevor Hastie and Robert Tibshirani. 2013. *An introduction to statistical learning*. Vol. 112 Springer.

Kleinberg, Jon, Jens Ludwig, Sendhil Mullainathan and Ziad Obermeyer. 2015. “Prediction policy problems.” *American Economic Review* 105(5):491–495.

Lentz, Erin C, Hope Michelson, Katherine Baylis and Yang Zhou. 2019. “A data-driven approach improves food insecurity crisis prediction.” *World Development* 122:399–409.

Lundberg, Scott M and Su-In Lee. 2017. “A unified approach to interpreting model predictions.” *Advances in neural information processing systems* 30.

Martini, Giulia, Alberto Bracci, Lorenzo Riches, Sejal Jaiswal, Matteo Corea, Jonathan Rivers, Arif Husain and Elisa Omodei. 2022. “Machine learning can guide food security efforts when primary data are not available.” *Nature Food* 3(9):716–728.

Meerza, Syed Imran Ali, Syed Irfan Ali Meerza and Afsana Ahamed. 2021. “Food Insecurity Through Machine Learning Lens: Identifying Vulnerable Households.”.

Mullainathan, Sendhil and Jann Spiess. 2017. “Machine learning: an applied econometric approach.” *Journal of Economic Perspectives* 31(2):87–106.

Nica-Avram, Georgiana, John Harvey, Gavin Smith, Andrew Smith and James Goulding. 2021. “Identifying food insecurity in food sharing networks via machine learning.” *Journal of Business Research* 131:469–484.

Okori, Washington and Joseph Obua. 2011. Machine learning classification technique for famine prediction. In *Proceedings of the world congress on engineering*. Vol. 2 pp. 4–9.

Raleigh, Clionadh, rew Linke, Håvard Hegre and Joakim Karlsen. 2010. “Introducing ACLED: An Armed Conflict Location and Event Dataset.” *Journal of Peace Research* 47(5):651–660.
URL: *https://doi.org/10.1177/0022343310378914*

Shapley, Lloyd S et al. 1953. “A value for n-person games.”.

van der Heijden, Wesley, Marc van den Homberg, Martijn Marijnis, Marijke de Graaff and Hennie Daniels. 2018. Combining open data and machine learning to predict food security in Ethiopia. In *2018 International Tech4Dev Conference: UNESCO Chair in Technologies for Development: Voices of the Global South*.

Von Luxburg, Ulrike and Bernhard Schölkopf. 2011. Statistical learning theory: Models, concepts, and results. In *Handbook of the History of Logic*. Vol. 10 Elsevier pp. 651–706.

Wasserman, Larry. 2004. *All of statistics: a concise course in statistical inference*. Vol. 26 Springer.

Weidmann, Nils B and Sebastian Schutte. 2017. "Using night light emissions for the prediction of local wealth." *Journal of Peace Research* 54(2):125–140.

Westerveld, Joris JL, Marc JC van den Homberg, Gabriela Guimarães Nobre, Dennis LJ van den Berg, Aklilu D Teklesadik and Sjoerd M Stuit. 2021. "Forecasting transitions in the state of food security with machine learning using transferable features." *Science of the Total Environment* 786:147366.

WFP, VAM. 2008. "Food consumption analysis: calculation and use of the food consumption score in food security analysis." *WFP: Rome, Italy* .

WFP, WHO, UNICEF et al. 2022. "The state of food security and nutrition in the world 2022.".

Yeh, Christopher, Anthony Perez, Anne Driscoll, George Azzari, Zhongyi Tang, David Lobell, Stefano Ermon and Marshall Burke. 2020. "Using publicly available satellite imagery and deep learning to understand economic well-being in Africa." *Nature communications* 11(1):2583.

Zhou, Yujun, Erin Lentz, Hope Michelson, Chungmann Kim and Kathy Baylis. 2022. "Machine learning for food security: Principles for transparency and usability." *Applied Economic Perspectives and Policy* 44(2):893–910.

Zhou, Yujun and Kathy Baylis. 2019. "Predict Food Security with Machine Learning: Application in Eastern Africa.".