Robust Statistical 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:
Dr. Clara Brandi, University of Bonn, IDOS
Dr. Lukas Kornher, University of Bonn, ZEF

Submitted in [month and year] by:
Gewei Cao
Matriculation Number: [number]

Acknowledgement

During my master's study at the University of Bonn, I obtained solid academic and practical training in Economics and data analysis. I hope this paper could show my knowledge and analytical skills in machine learning, data management, and economics. Without the help of my supervisors Dr. Clara Brandi, *University of Bonn, IDOS*, and Dr. Lukas Kornher, *University of Bonn, ZEF*, this paper would not be finished. They inspired me and guided me in the research field of food security. They also gave me precious suggestions for academic writing. When I was working as a Research Assistant in ZEF with Dr. Lukas Kornher, he suggested and inspired me to write my master's thesis in the field of food security and to use open-sourced data with micro survey data. I appreciate the experience of Research Assistance in ZEF, which gave me opportunities to develop my data skills and show my capability, besides, this experience also supply me with the background knowledge needed for this thesis. Meanwhile, thanks to Mr. Emmanuel who provides the Uganda survey data. In addition, thanks to those open-source databases, I always believe open-source data, books, and papers will encourage more people to participate in academia and the creation and spreading of knowledge, which will make this world better. Hope this paper will help the people who are suffering from hunger and poverty.

In addition, also thank my friends and colleagues, who give me advice during my master's study, in the field of machine learning and computer science. Last but not least, thank my girlfriend who supports me during my master's study, and supports my further Ph.D. study in Germany.

Finally, many thanks to the Economics department of Rheinische Friedrich-Wilhelms-Universität Bonn, which provides solid and various economics training and education, and also leads me into the broad field of Economics academia.

Contents

Abstract

The food insecurity issue in eastern Africa and Uganda is serious, facing the shocks such as the Covid pandemic and war in Ukraine, a robust ML 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. It is shown that XGBoost and random forests behave robustly when facing the shock of Covid. The monthly average AUC of XGBoost and random forests are 0.83, and with the model default prediction, an accuracy of around 80% and recall of around 72% are achieved.

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, 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. Meanwhile, the war in Ukraine hindered the recovery from the pandemic and disrupted the international food and energy market, pushing inflation higher (IMF (2022)). Due to this conflict, low-income food-deficit countries that rely on the import of food and fertilizers will suffer more from the high price of these kinds of commodities (WFP, UNICEF et al. (2022)).

According to the definition of the Food and Agricultural Organization of the United Nations (FAO), a person is food insecure when they lack regular access to enough safe and nutritious food for normal growth and development and an active and healthy life. In the year 2021, there were approximately 768 million people in this world facing hunger (undernourishment) (WFP, UNICEF et al. (2022)). Therefore, food insecurity is a crucial humanitarian and development challenge for the world.

From the perspective of the economy, food insecurity negatively affects economic development in several ways. Firstly, it can raise the risk of poor health outcomes such as malnutrition, obesity, and stunting, which can hinder the physical and cognitive development of children (e.g. WFP, UNICEF et al. (2022) Kang et al. (2018), Guerrant et al. (2008)). Food insecurity also increases the risk of chronic diseases (e.g. Seligman, Laraia and Kushel (2010),Weaver, Fasel et al. (2018),Laraia (2013)). Above both will reduce the productivity of the affected population, and increase the cost of health expenditure of the society.

In addition, food insecurity can contribute to social and political instability. For example, food insecurity can perpetuate violent conflicts, and contribute to a vicious cycle (Brinkman and Hendrix (2011), Hendrix and Brinkman (2013)). Food insecurity is also an important determinant of migration intentions and preparations (Smith and Floro (2020)). Furthermore, during Covid-19, the increased food insecurity may increase the migration pressure (Smith and Floro (2020)).

Food insecurity is apparently a critical humanitarian and development issue, particularly in Africa. According to WFP, UNICEF et al. (2022), in the year 2021, more than one-third of people in hunger are in Africa, the prevalence of undernourishment in Africa is 20.2%, and in Eastern Africa, it is 29.8%, in other words, there are around 136.4 mil-

lion of people suffered undernourishment. Meanwhile, according to FAOSTAT, in 2021, the prevalence of severe food insecurity in Eastern Africa is 28.7%, and the prevalence of severe or moderate food insecurity is 66.9%. Africa, especially Eastern Africa, is facing a pressing food security challenge. The United Nations World Food Programme (WFP), Action Against Hunger, and many other humanitarian projects are concerning food insecurity in Africa. These projects need to identify or forecast food insecurity effectively, in order to make decisions and take action.

Moreover, 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 more 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, under the shocks of Covid-19 and the war in Ukraine, the robustness of the ML model for food insecurity prediction need to be considered. 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 whether exists a relatively robust ML algorithm when facing such shock and trying to find a high-performance ML model for the predicting task in Uganda.

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

2 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

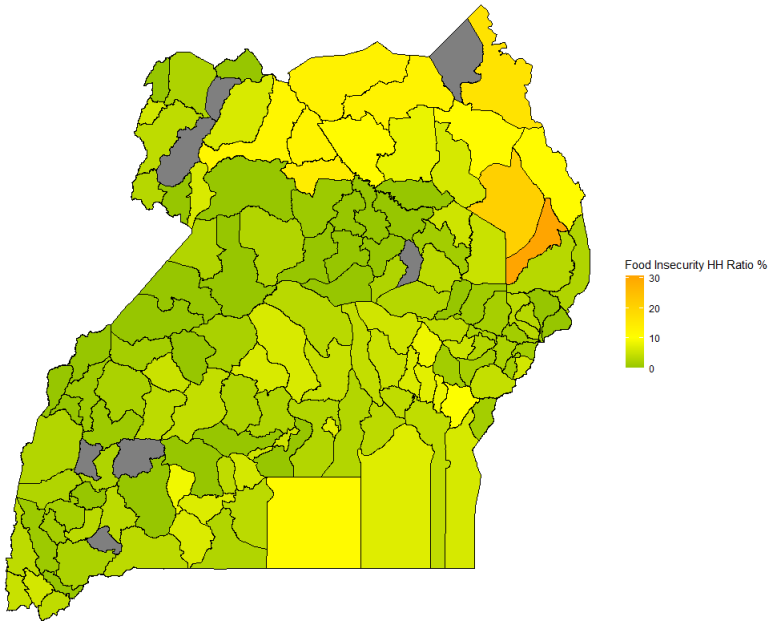


Figure 1: Uganda Food Insecurity Map of UNHS 2019/20

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 1 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.

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.

3 Statistical Learning and Machine Learning

3.1 Introduction

Statistical learning refers to a set of tools for making sense of complex datasets (James et al. (2013)), and it is closely related to the concept and practice of Machine Learning (ML). 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 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. OLS), 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, we only focus on supervised learning.

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 the best parameters or functions we want. Note that we assume there are p features, i.e. $x_i \in R^p$ 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 the machine learning context.

In this paper, food security 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. We call a food insecure observation a positive case and a food secure observation a negative case in the following table.

		True classes		
		Positive	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

In addition, 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 the probability. The closer the curve to the top left corner, the better the model (see Figure (6) in the later section). Because this means there exists a threshold with low FPR and TPR. To measure such a pattern, the area under curve (AUC) is a good measure. The higher AUC, the more effective the model. A value of 1 for AUC means perfect fitting, which implies an overfitted model, and a value

of 0.5 for AUC implies the model performs the same as a random guess.

The performances of different algorithms are compared 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.

Finally, 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. The goal of this study is to find the most robust and precise algorithm.

3.2 ML working flow

3.2.1 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 UNHS 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 4. In addition, some open-source data have different district names, the district names in UNHS are taken as a benchmark, and match those differences manually or solve by a Uganda national scale mean imputation. After data management, 13,302 households are obtained as our samples.

3.2.2 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 introduced 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, 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, this has a risk of data leakage. The training and testing data are split monthly in this study.

3.2.3 Problem of imbalanced data

Food insecurity data here is imbalanced, the proportion of food severely insecure ("Poor" class) 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

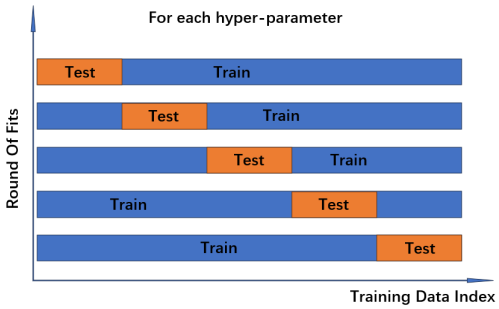


Figure 2: Visualization of Cross Validation

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, we need the sample size to be $N_{10} = 100^{10}$ with 10 predictors (Hastie et al. (2009)). Therefore, before resampling, we have to reduce the dimensionality of our input features. The ML result section will show the resampling performances.

3.2.4 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 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 type I and type II errors, see the ML result section for the reason. 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 2 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 label, 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 2.

3.2.5 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 firstly by AUC ROC, and we will inspect the dynamic of accuracy, recall, precision, and f1 score for different decision threshold probabilities.

3.3 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 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.

3.4 Support Vector Machine

Support Vector Machine (SVM) is a very famous ML algorithm. It provides a nice intuition for the classification problem. Data X locate at a p dimensional real number space, 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

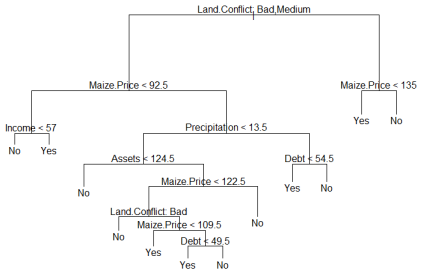


Figure 3: Example of a single decision tree

is used, which allows some training points on the wrong side of the hyperplane or in the margin. The optimization problem of the support vector classifier is

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

, 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.

3.5 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 3 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 functions or some criterions, 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 it to control the number of terminal nodes.

However, the above decision tree algorithm is still non-robust and has high variance. Thus, we introduce techniques of bagging and boosting. 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, we only update our $\hat{f}(x)$ at a little learning rate λ, for example, $\lambda = 0.01$. Then we 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, we modify our data, in other words, we fit the new "residual" each round through boosting. Nevertheless, in the context of classification, instead of "residual" in regression context, the $log(odds) = log(\frac{p}{1-p})$ is usually used in each round of fitting.

3.5.1 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, we can reduce the variance of function, and grow decorrelated trees.

To be specific, we draw a boostrap sample of size N and use m out of p features (predictors) to grow a tree T_b, repeat B times, and we have a set of trees $\{T_b\}_1^B$. Then we average the prediction result of $\{T_b\}_1^B$ to make the final prediction, we can take the mean of single predictions in the regression case and make a majority vote in the classification case.

3.5.2 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, we need a regularizer for pruning and controlling the scale of predictions by $\Omega(f_b) = \gamma|T| + \frac{1}{2}\lambda \sum_{j=1}^{|T|} f_j^b(x)^2$. Then, we can 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, we 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.

3.6 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, we introduce SHAP (SHapley Additive exPlanations) proposed by Lundberg and Lee (2017). 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 $\hat{f_X}$. The Shapley value of most ML models could be solved by SHAP.

SHAP enables us 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.

4 Data of Uganda

Uganda National Household Survey (UNHS) is undertaken by the Uganda Bureau of Statistics. Our food security indicator (response variable) and some of predictors are provided via UNHS. The UNHS of 2019/20 and 2016/17 data are used. In the 2016/17 data, there are 15,912 households were interviewed, and in the 2019/20 data, there were 15,659 households. The 2016/17 data covers from July 2016 to June 2017, consecutively 12 months. Besides, 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.

4.1 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))

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 cumulative mean food consumption frequencies for 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. The case for 2016/17 UNHS can be found in the appendix.

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 our 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.

Figure 4: cumulative mean consumption frequency 2019/20

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

In summary, the 2019/20 FCS of 13,302 households are calculated, the FCS distribution of UNHS is shown in Figure 5, and the distribution of 2016/17 FCS data can be found in the appendix. Figure 5 shows that most households are in the "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.

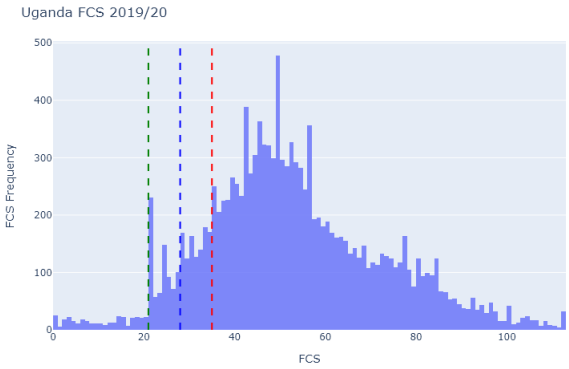


Figure 5: Distribution of FCS in Uganda 2019/20

4.2 Predictors

4.2.1 Open source data

Nightlight data could be 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[1] (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[2] 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.

The climate variables 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.*[3] They are extracted with the map data as the nightlight section 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 data is also used, it is provided by VAM Food Security Analysis[4], and for those districts without NDVI data, an imputation of Uganda national scale data happens.

The conflict data are obtained at *Armed Conflict Location & Event Data Project (ACLED),*[5] 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, I generated the total fatalities for each month for all kinds of conflicts, 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 (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[6]. For those districts without NDVI data, an imputation of Uganda national scale data happens. Lagged data are added.

[1] https://eogdata.mines.edu/products/vnl/
[2] https://data.humdata.org/
[3] https://power.larc.nasa.gov/data-access-viewer/
[4] https://dataviz.vam.wfp.org/seasonal_explorer/rainfall_vegetation/visualizations
[5] www.acleddata.com
[6] https://dataviz.vam.wfp.org/seasonal_explorer/rainfall_vegetation/visualizations

4.2.2 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. A detailed variable list could be found in the appendix. 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 distribution of some features conditional 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 insecurity ratio of Uganda 2016/17 is chosen as a feature from the UNHS 2016/17 data. 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.

4.2.3 Final features

After the data management and feature engineering, there are 50 features selected to train ML models. The correlation coefficients matrix of continuous variables is shown in the Appendix. Some open-source macro-level data are also manipulated, such as using the mean value of the previous four months' data and their logarithm transformation as features.

5 ML Models and Results

5.1 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 series data and future predicting will not be considered, only the present data will be input to 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 our prediction model in this food security context. To evaluate the robustness of ML models, we should either cover the shock of Covid in the training data or cover the shock in the testing data, since we assume that in UNHS data there is a distributional 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 the training data. This ex-post model provides us the 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 we can define training data and testing data:

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

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 next 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 an accidental shock.

5.2 ex-post robust design

5.2.1 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 $\leq$ 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 (6) 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.

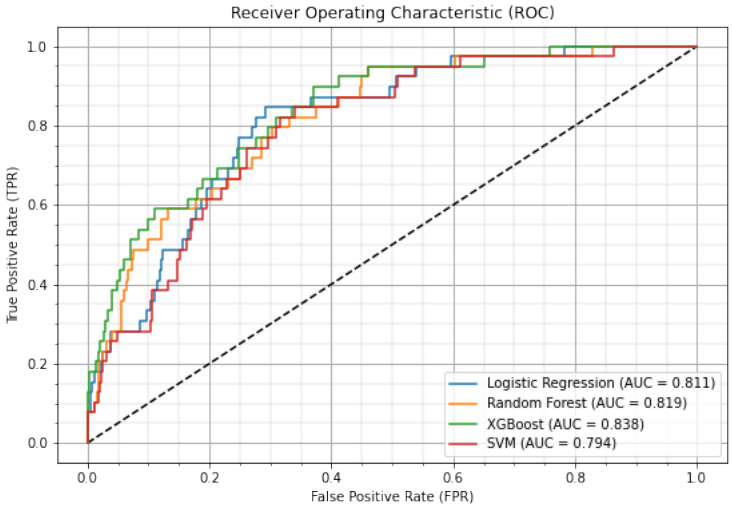


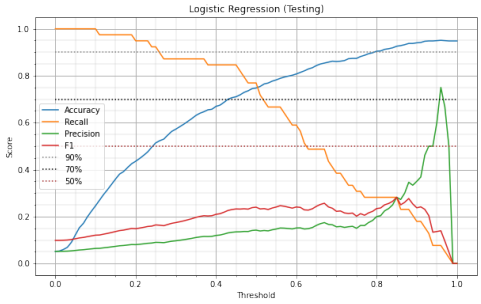
Figure 6: ROC for Testing Data

In practice, depends on the welfare v.s. cost trade-off, one can adjust the decision threshold to a probability different from 0.5, figure (7) shows the metrics trade-off. The model

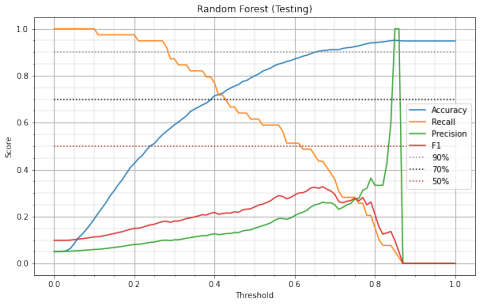
Models	accuracy	precision	recall	f1	AUC
Logistic	0.748	0.142	0.769	0.240	0.811
Random Forest	0.799	0.150	0.615	0.241	0.819
XGBoost	0.818	0.164	0.615	0.259	0.838
SVM	0.744	0.130	0.692	0.219	0.794

Table 3: Model performance for default output

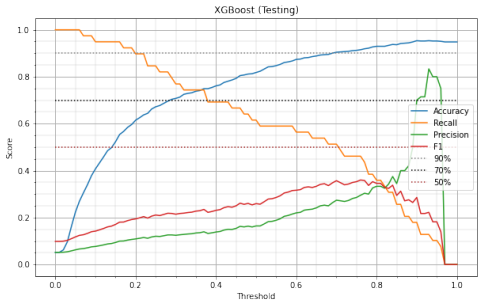
accuracy and recall curves of SVM (Figure 7.d) change sharply with respect to the threshold probability of less than 0.1, this gives the decision maker a narrow space to adjust the threshold and trade-off between welfare and cost. The algorithm of SVM targets the best separation hyperplane instead of 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 7.a, 7.b and 7.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 (7) 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.



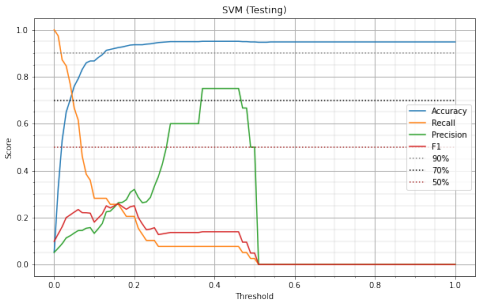
(a) Logistic Regression



(b) Random Forest



(c) XGBoost



(d) Support Vector Machine

Figure 7: Performances with different threshold

Therefore, we can conclude that tree-based models and regularized logistic regression are more robust against a shock, and the XGBoost outperforms the others. SVM cannot reveal the complex interaction among so many predictors. Regularized logistic regression benefits from the feature selection $l1$ regularization, and performs almost as robustly as

random forest. Meanwhile, ensemble or boosting tree models such as random forest or XGBoost can capture those interactions and shock dynamics. Last but not least, XGBoost is a regularized gradient boosting method, so it could better overcome over-fitting and performs more robustly to a shock in given training data.

5.2.2 SHAP interpretation

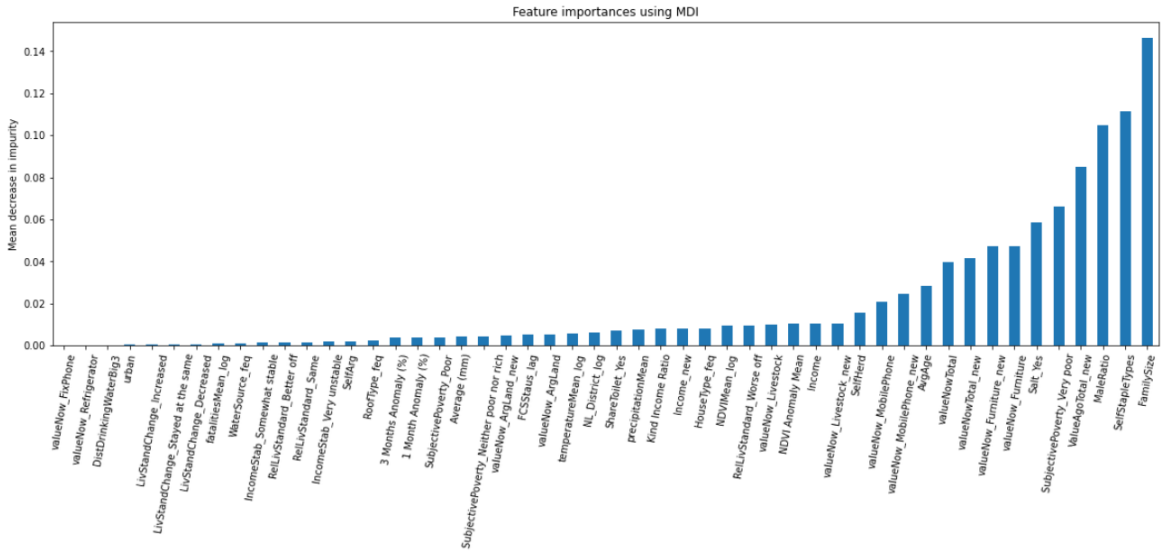
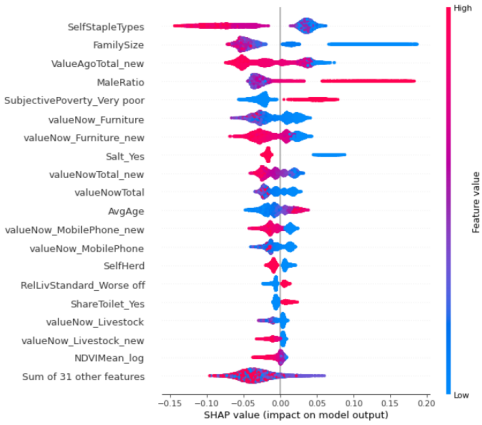


Figure 8: Feature Importance of Random Forest

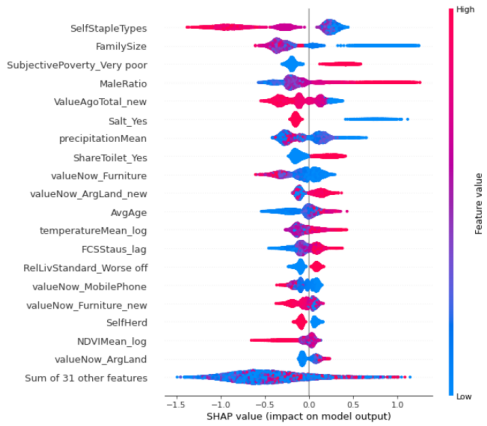
The SHAP method and its Python package enable us to inspect the feature contributions of food insecurity prediction in our microdata. Meanwhile, for tree-based models, feature importance scores can be computed, it is calculated by the contributions of each predictor when growing decision trees. However, the caveat of SHAP and feature importance score is that they do not imply any causality.

Figure (8) 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.

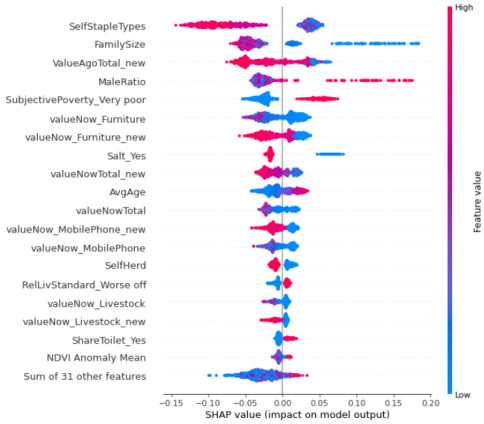
However, the feature importance score does not reveal how each feature influences the prediction. SHAP value is helpful in exploring greater details about ML models. For our food insecurity problem, a higher SHAP value for an observation brings a higher probability of being food insecure (class 1). Figure (9) shows the top 20 influential features of our tree-based models. Random forest (9.a) explores the most in the household-level data, and the top features are similar to the result in Figure (8). 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 cold explain why the AUC of logistic regression and random forest are close, because logistic regression may explore this data more. Furthermore, we can use SHAP to understand the impact of each feature, similar to how we understand the coefficients of OLS linear regression. The top two of both



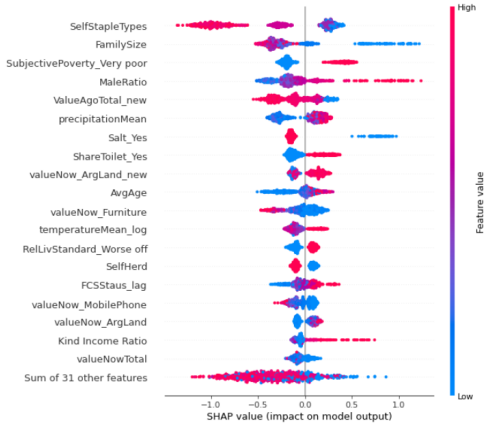
(a) Random Forest (training data)



(b) XGBoost (training data)



(c) Random Forest (testing data)



(d) XGBoost (testing data)

Figure 9: Top 20 SHAP Value for tree-based models

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 (9).c and (9).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 (10). 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 (9).

Finally, SHAP is also inspiring in the feature selection step. After training the model, one can use the SHAP plot as the training part of Figure (9), and drop the less influential features. By doing this feature selection, the noise and dimensionality of data are reduced, and the model performance could be improved. Nevertheless, to prevent data leakage,

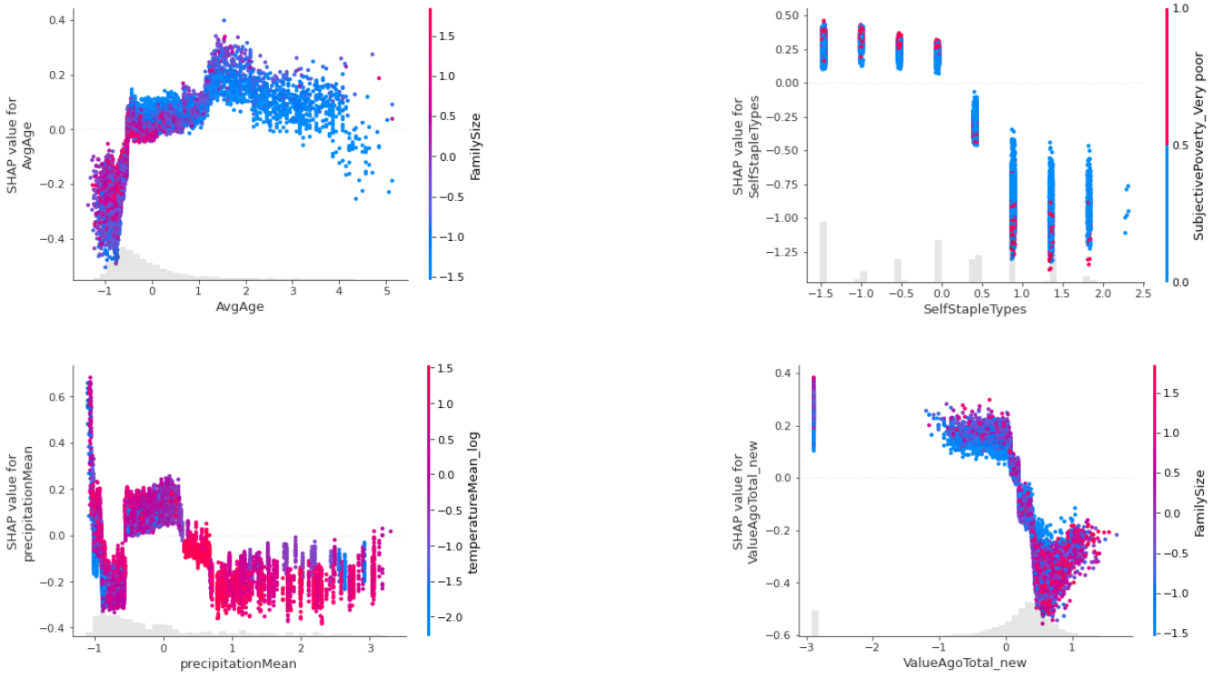


Figure 10: Scatter Plot of variable interactions

only the SHAP information of the training data should be used.

5.2.3 Identifying borderline food insecurity

In addition to identifying the "Poor" food security households, we here try 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 and logistic regression still outperform. 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. Our 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.

5.2.4 Resampling methods

As mentioned in Section 3, 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 (11) there is no strong correlation within 10 features. Notably, Although resampling with fewer features might improve the sensitivity of SVM, the inflexibility of SVM leads it to be dropped. Figure (12) 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 give tree limited information of the minority class. The improvement

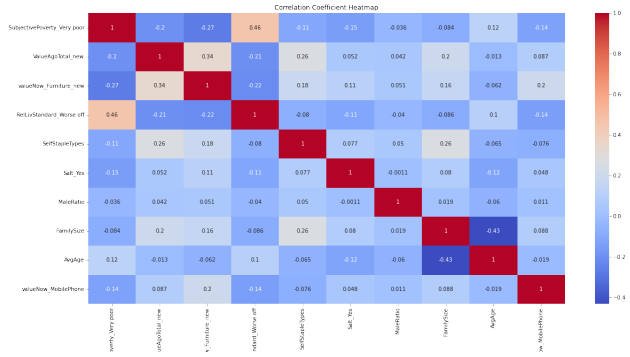


Figure 11: Correlation Coefficients of top 10 features

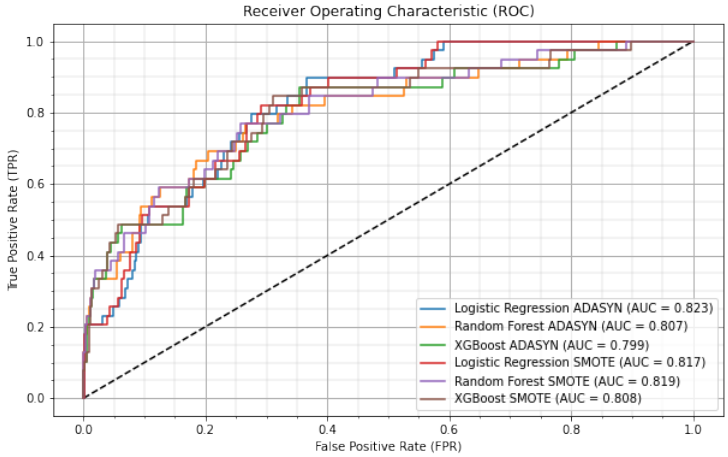


Figure 12: ROC and AUC for two resampling techniques

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.

5.3 ex-ante robust design

5.3.1 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 13 shows the AUC of each model and the overtime AUC mean and standard deviation. Firstly, the means and the standard deviations of ACU for the four models are similar. Besides, XGBoost has the highest standard deviation. XGBoost gives a relatively higher AUC compared with others during the Covid, but performs worse than others be-

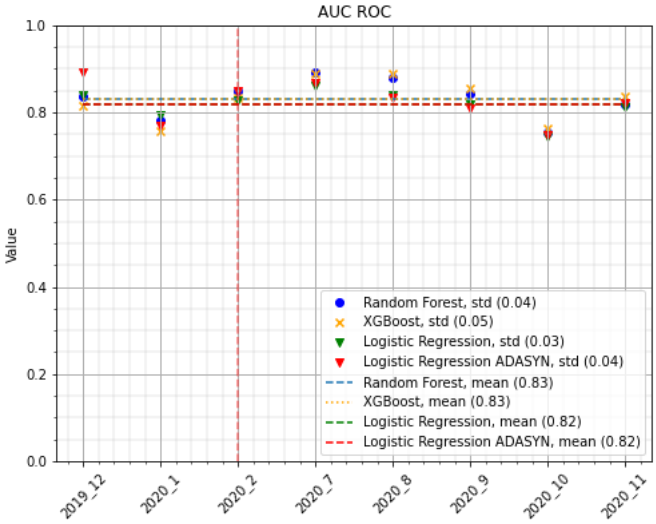
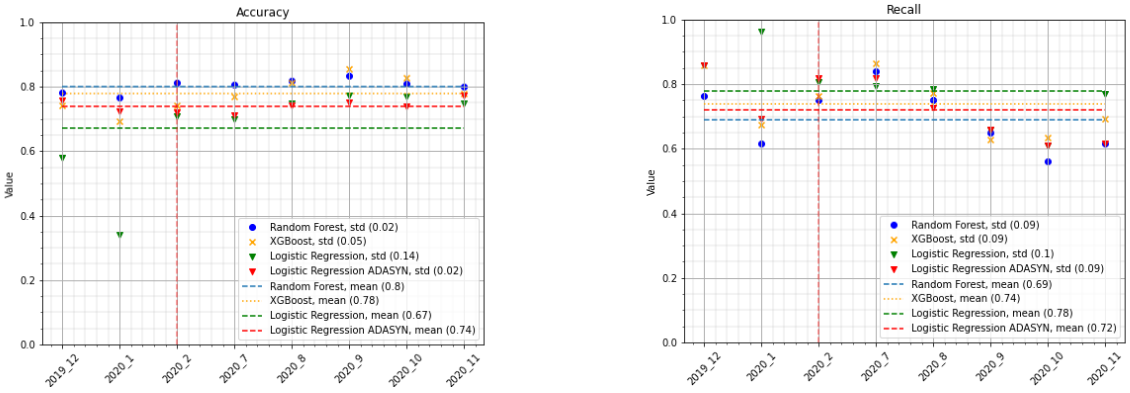


Figure 13: AUC



(a) Accuracy (b) Recall

Figure 14: Default output of accuracy and recall

fore the Covid breakout, this could explain why XGBoost has a relatively higher overtime standard deviation. 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 2020, which is the month of the breakout of Covid in UNHS data, the two tree-based models 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 in the training data.

In addition, Figure 14 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. As the AUC results, the accuracy performances of tree-based models are more robust. Although the recall performances of logistic models are over the tree-based models, a slightly higher AUC of

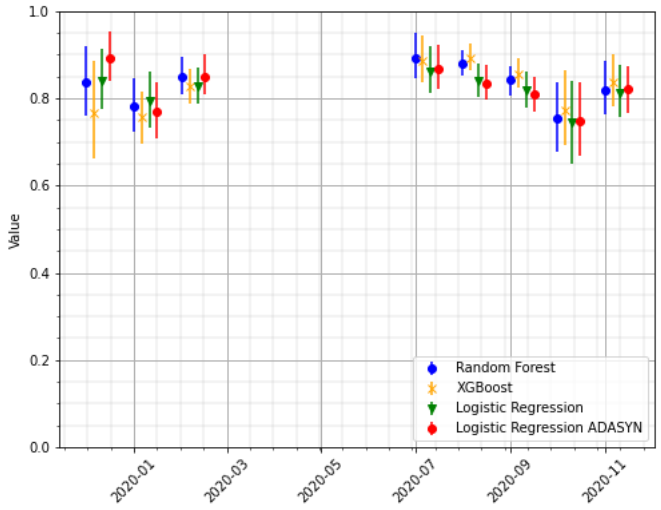


Figure 15: AUC with bootstrap

tree-based models enables them to be adjusted.

5.3.2 Bootstrap robustness

Above we only use 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 whole monthly testing data set with replacement totally B times, $B = 500$, and the trained model fits 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 15 shows the bootstrapped 95% confidence interval of AUC, the points of monthly AUC are the same as in Figure 13. The confidence interval is the pivotal interval (more to see Wasserman (2004)). 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 15, 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 shrink. Tree-based models perform better facing the shock. 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 16 shows the standard deviation of bootstrapped AUC for each month, and the overall standard deviation on the label is computed by all bootstrap samples and all months. Although the overall standard deviation of logistic regression with the ADASYN sample is higher than without it, 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

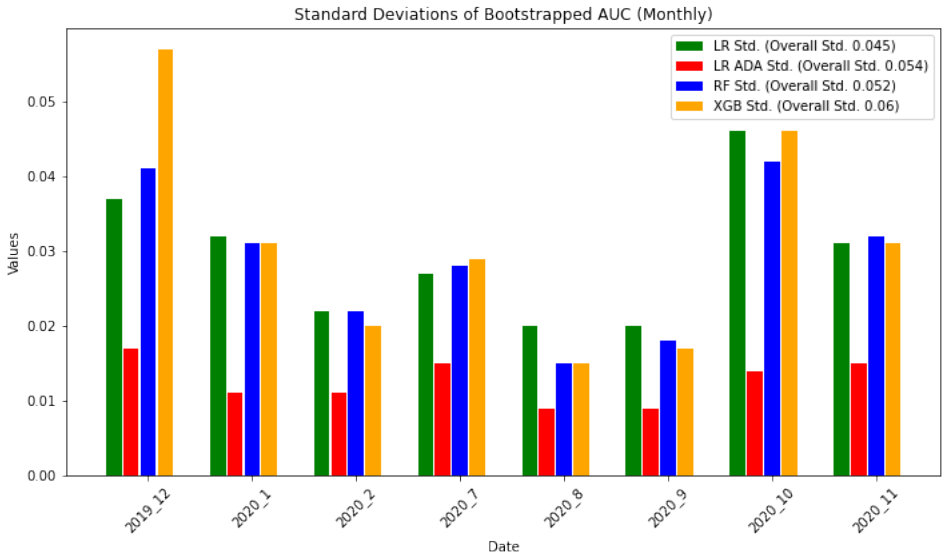
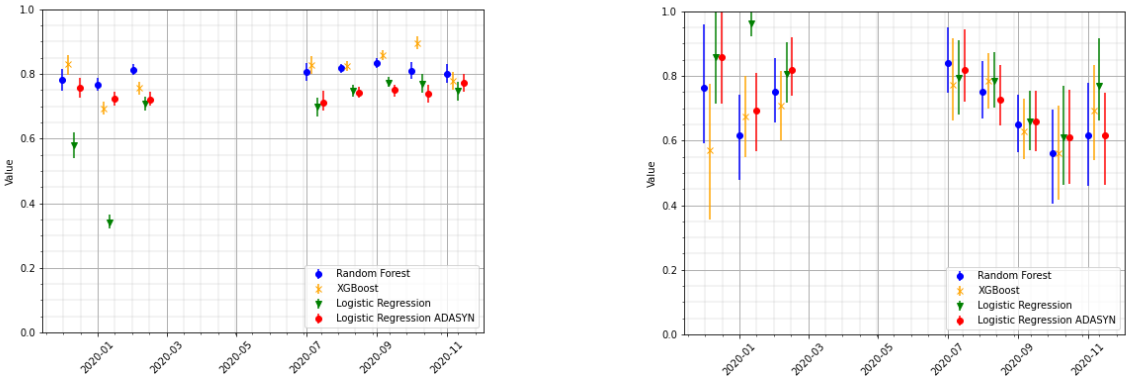


Figure 16: Standard deviation of bootstrapped AUC



(a) Accuracy (b) Recall

Figure 17: 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. This bootstrapped standard deviation implies the reliability of the output metrics for testing data, however, the distribution of estimated AUC and the influence of bootstrap should be considered, this study will just take the more bootstrap ad hoc pivotal confidence interval into model evaluation.

Figure 17 shows the default output accuracy and recall with 95% confidence interval. As shown in the ex-post model, tree-based methods perform not so well 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 Conclusion

In summary, Uganda is facing the threat of food insecurity, and the breakout of the Covid pandemic and war in Ukraine requires a robust ML for food insecurity forecasting. This study explores the UNHS data and the open-sourced data to find robust ML models.

Two model designs are introduced, ex-post model design considers the shock in history and tries to find robust models after or during the shock. The ex-ante design tries to find robust models when encountering shock. With the ex-post model design, tree-based models and regularized logistic regression outperform, and the UNHS household data contributes the most to prediction. Meanwhile, resampling techniques do not help for tree-based models, but logistic regression is improved. In the ex-ante model design, which is a more practical design, the tree-based models outperform and show good robustness. We can conclude that for robust ML model selection, tree-based models such as random forest and XGBoost are preferred, because decision trees could discover more interactions and behave robustly facing or after a shock.

Appendix

Figures

Figure 18: cumulative mean consumption frequency 2016/17

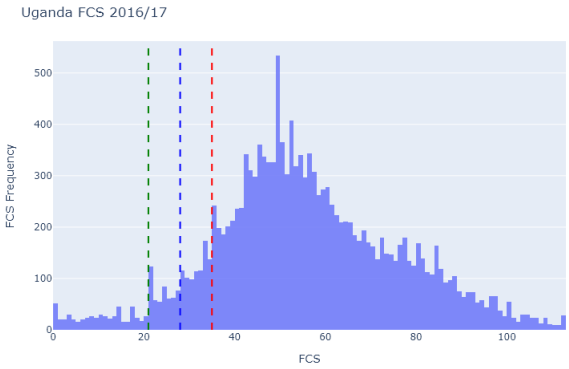


Figure 19: Distribution of FCS in Uganda 2016/17 (15144 obs.)

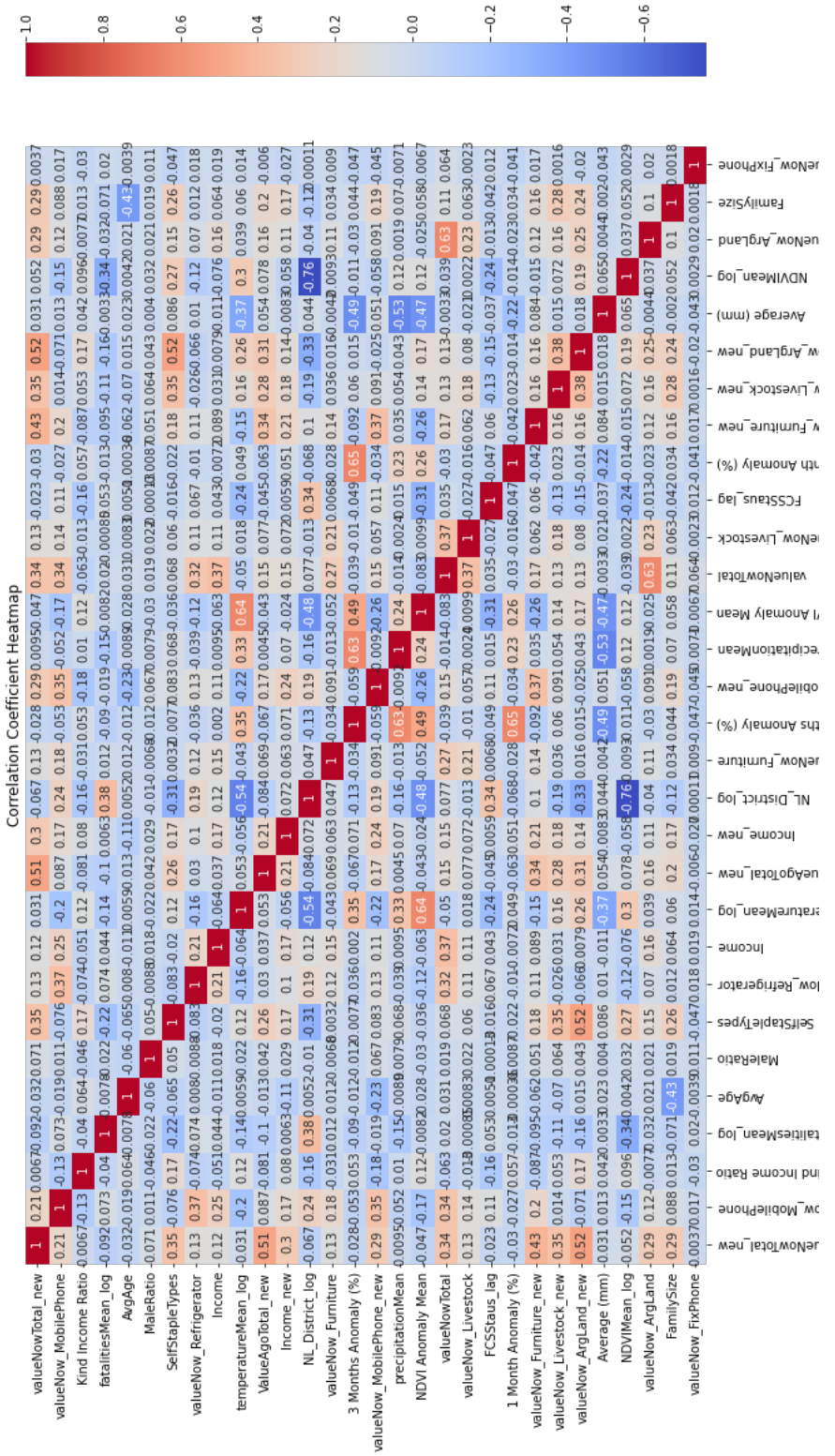


Figure 20: Correlation Coefficient of all continuous features

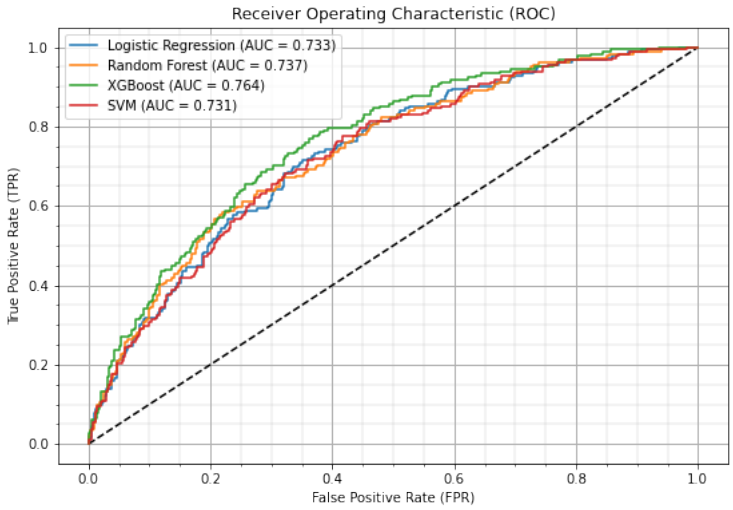
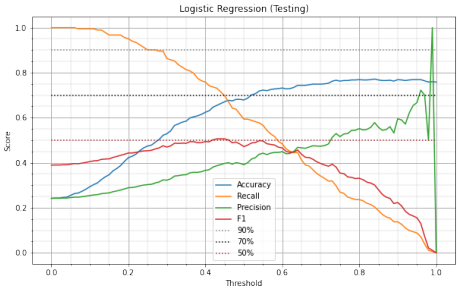
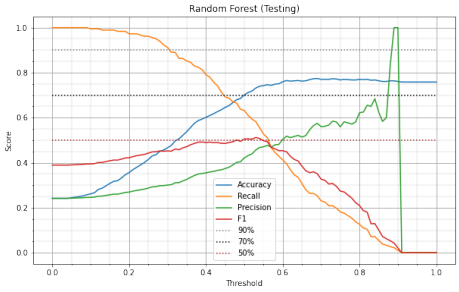


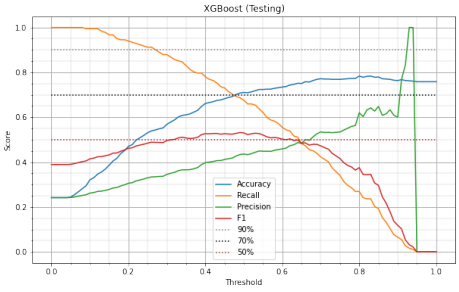
Figure 21: ROC for Testing Data (FCS = 35 as threshold)



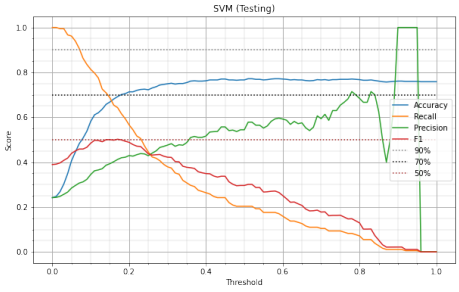
(a) Logistic Regression



(b) Random Forest



(c) XGBoost



(d) Support Vector Machine

Figure 22: Performances with different threshold (FCS = 35 as threshold)

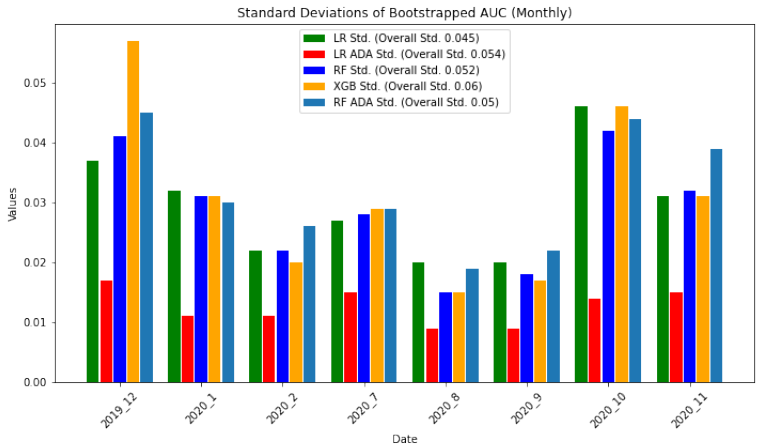


Figure 23: Standard deviation of bootstrapped AUC with ADASYN RF

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

Brinkman, Henk-Jan and Cullen S Hendrix. 2011. "Food Insecurity and Violent Conflict: Causes." *Consequences, and Addressing the Challenges, World Food Prgramme* .

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.

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.

Guerrant, Richard L, Reinaldo B Oriá, Sean R Moore, Mônica OB Oriá and Aldo AM Lima. 2008. "Malnutrition as an enteric infectious disease with long-term effects on child development." *Nutrition reviews* 66(9):487–505.

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.

Hendrix, Cullen and Henk-Jan Brinkman. 2013. "Food insecurity and conflict dynamics: Causal linkages and complex feedbacks." *Stability: International Journal of Security and Development* 2(2).

IMF. 2022. "World Economic Outlook: War Sets Back the Global Recovery." *IMF* .

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

Kang, Yunhee, Víctor M Aguayo, Rebecca K Campbell and Keith P West Jr. 2018. "Association between stunting and early childhood development among children aged 36–59 months in South Asia." *Maternal & Child Nutrition* 14:e12684.

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

Laraia, Barbara A. 2013. "Food insecurity and chronic disease." *Advances in Nutrition* 4(2):203–212.

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.

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

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*

Seligman, Hilary K, Barbara A Laraia and Margot B Kushel. 2010. "Food insecurity is associated with chronic disease among low-income NHANES participants." *The Journal of nutrition* 140(2):304–310.

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

Smith, Michael D and Maria S Floro. 2020. "Food insecurity, gender, and international migration in low-and middle-income countries." *Food Policy* 91:101837.

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.

Weaver, Lesley Jo, Connor B Fasel et al. 2018. "A systematic review of the literature on the relationships between chronic diseases and food insecurity." *Food and Nutrition Sciences* 9(05):519.

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.

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.