Modeling Food and Housing Insecurity at UTEP - Phase 1

Analysis Plan

George E. Quaye Johnson Koomson Willliam O. Agyapong

University of Texas, El Paso (UTEP)

Department of Mathematical Sciences

Contents

1 Introduction **3**

1.1 Data Description 3

2 Data Preprocessing **3**

2.1 Rules for Defining Response Variables 3

2.1.1 A measure of Food Insecurity 3

2.1.2 A measure of Housing Insecurity 3

3 Exploratory Data Analysis **3**

3.1 Missing Data/Outliers and their treatments 4

4 Modeling Approaches **4**

4.1 Data Partitioning 4

4.2 Predictive Models 4

4.2.1 Logistic Regression 5

4.2.2 K-Nearest Neighbors Classifier 5

4.2.3 Classification Trees 5

4.2.3.1 Random forest 6

4.2.4 Support Vector Machines (SVM) 6

4.2.5 Naive Bayes Classifier 6

4.2.6 Multivariate Adaptive Regression Splines . . . 7
4.2.7 Artificial Neural Network . . . 7
4.3 Model Validation . . . 7
4.4 Performance/Evaluation Metrics . . . 7
4.4.1 F-beta Score (F_β) . . . 7
4.4.2 Area Under the Receiver Operator Curve (AUROC) . . . 8
4.4.3 Area Under the Precision Recall Curve (AUC-PR) . . . 8
4.5 Model Selection . . . 9
4.6 Model Assessment . . . 9

5 Model Deployment **9**

References **9**

This document serves to provide a road map to the approaches our group intend to adopt and execute in terms of our modeling approaches and general outline of our final report. We, however, remark that the various plans set forth are subject to changes according to recommendations and suggestions received from our domain experts (Dr. Amy Wagler and her research colleagues on this same project) upon consultations, and any other future modifications that we shall consider appropriate.

1 Introduction

This section will primarily provide a background to the study by means of an overview of food insecurity and housing insecurity in the US in general and at El Paso, the satellite region of our target population , in particular. It will also appeal to literature covering already existing modeling aproaches, if any.

1.1 Data Description

2 Data Preprocessing

2.1 Rules for Defining Response Variables

2.1.1 A measure of Food Insecurity

2.1.2 A measure of Housing Insecurity

3 Exploratory Data Analysis

In this section, we will provide descriptive analyses of the data including numerical summaries and graphical displays as a first-hand insight into the data to help ourselves and our audiences to better understand the underlying dataset. We will primarily be looking at the distributions of our target variable (death event) and all the predictor variables.

In terms of the types of graphs or visualizations, I would like my group to consider creating comparative boxplots/histograms, segmented or comparative bar graphs and, possibly, density plots. We will critically assess these graphs and settle on a few which will help us to effectively tell a compelling story about the data. We will build animation effects into some of the graphs where possible or appropriate.

- **Cross Tabulations (Contingency Tables)**: Provide numerical summary of how responses differ by some categories.
- **Comparative Boxplots/Histograms/Density plots**: These graphs will be used to describe the distribution of the continuous or numeric variables across the levels of the target variable.

- **Bar graphs (Segmented/Comparative bar graphs) and Mossaic Plots**: To explore how the available categorical predictors are distributed across the levels of the target or response variable.
- **Correlation plots** will be utilized to detect the presence of multicolinearity among the continuous predictors.

3.1 Missing Data/Outliers and their treatments

The available data will be inspected for missing and unusual values. Any missing data and/or outlying data identified from our data preprocessing and exploratory analysis will be treated appropriately depending on the nature of the missingness and the information we are able to gather about the possible reasons for the missingness or unusual values. Possible methods to likely to be considered include imputation, listwise deletion, among others.

4 Modeling Approaches

4.1 Data Partitioning

For the purpose of predictions or testing on the machine learning algorithm models or estimation of the evaluation metrics for the best model selection, a division of the entire into say training data and testing data will be carried out. The training data would be used to fit the models and the testing data was used to derive the metrics for model selection or testing. This partitioning will be done by using the strategy of V -fold cross-validation on the misclassification errors, with V = n, where n would specified base on the sample size of the entire data set. We will randomly divide the data into V folds with stratification on the target variable category. This assures that similar proportions of 0s and 1s are preserved in each fold, and set.seed() will be used for easily reproducible results.

4.2 Predictive Models

In this part of the analysis, since the nature of the tasks present a classification problem, we will fit several classification models to the target variable as a function of all the possible predictors in the dataset that we will deem fit. All the models considered are *supervised* learning algorithms.

Candidate models include Logistic regression, Linear Discriminant Analysis (LDA), K-nearest neighbors (KNN), Support Vector Machines/Classifiers, Naive Bayes' algorithm, Multivariate Adaptive Regression Splines, Artificial Neural Network, and Tree-based models such as Random Forest, Bagging, or Boosting. Due to the sensitivity of the KNN model to different scales, the continuous variables will be normalized and kept across the other models for consistency, that is, if we end up fiting a KNN model.

4.2.1 Logistic Regression

Logistic Regression is very simple and one of the mostly used traditional machine learning algorithms. It is a statistical model for predicting binary classes. However, as remarked by James, Witten, Hastie, and Tibshirani (2013), multiple-class extension of Logistic Regression exists but *Discriminant Analysis* models are commonly used in place of it. The dependent variable in here follows Bernoulli distribution. Logistic Regression model uses the logistic function or sigmoid function for predictive modeling of the given problem, which takes value between 0 and 1. 1 will be predicted if the curve goes to positive infinity and 0 if it goes to negative infinity. The logistic Regression model performs the predictive analysis based on the relationship between the binary dependent variable and the other one or more independent variables from the given dataset.

Binary Logistic Regression, we will fit a regularized logistic regression model such as LASSO as one of the baseline classifiers for comparison. LASSO does not require the target variable to be normally distributed, no homogeneity of variance assumption is required, and with the aid of graph and output interpretation is effective under LASSO. LASSO, however, requires more data to achieve stability and is effective mostly on linearly separable.

4.2.2 K-Nearest Neighbors Classifier

K-nearest neighbors regression (KNN regression) is one of the simplest and best-known non-parametric methods. Unlike the Logistic Regression, no assumptions are made about the shape of the decision boundary. Therefore, we can expect this approach to dominate Logistic Regression when the decision boundary is highly non-linear. It uses a distance metric such as the Euclidean distance for separation between points in the feature space. In the KNN classification model, the prediction is purely based on neighbor data values without any assumption on the dataset. The K in the name of the model represents the number of nearest neighbor data values. This parameter can be tuned to find an optimal value. Based on K, the decision is made by the KNN algorithm on classifying the given dataset.

4.2.3 Classification Trees

According to (James et al., 2013), for a classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. In interpreting the results of a classification tree, we are often interested not only in the class prediction corresponding to a particular terminal node region, but also in the class proportions among the training observations that fall into that region. A recursive binary splitting is used to grow a classification tree. Classification trees are often preferred to other classifiers for the following reasons. One thing, they are non-linear classifiers which do not require the data to be linearly separable. Two, they are easy to read and very easy to explain to people. After a model is generated, it's easy to report back to others regarding how the tree works. Moreover, trees can be displayed graphically, and are easily interpreted

even bya non-expert (especially if they are small). Also, decision trees can easily handle qualitative predictors. However, trees can be very non-robust. In other words, a small change in the data can cause a large change in the final estimated tree. For this reason, we will use tree models that help circumvent this challenge and greatly improves the predictive performance by aggregating many decision trees. Such tree models include Random Forest, Boosting, and Bagging.

4.2.3.1 Random forest The random forest is a great predictive model for high dimensional data, and we will use RF as one of the baseline classifiers model. RF has high performance and accuracy with regards to modeling, provides feature importance of estimates, and can automatically handle missing values with no scaling required. However, given that RF has some advantages over other classifiers, its prediction time is high, can overfit the data, and also require more computational time and resources.

4.2.4 Support Vector Machines (SVM)

Support Vector Machines; SVM is a great classification machine learning model for classifying data of clear classes. Also does not require a predetermined cutoff point for prediction. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space so as to maximize the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall (Wikipedia, 2021). We will consider using this classification model because it is versatile and effective in high dimensional spaces. They are versatile in the sense that different Kernel functions can be specified for the decision function. Common kernels include polynomial kernels and radial kernels. Support vector classifiers are also able to address the problem of possibly non-linear boundaries between classes. We will try at least three kernel SVM for this analysis noticeably, the Linear SVM and two Non-Linear kernels SVM such as the Gaussian radial basis function (RBF) and Polynomial kernel SVM through packages such as caret and kernlab.

4.2.5 Naive Bayes Classifier

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It works on Bayes theorem of probability to predict the class of unknown data sets. Naive Bayes algorithm is particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. When the assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data. It

should be however noted that this assumption of independent predictors may be unrealistic since, in real life, it is almost impossible that we get a set of predictors which are completely independent. Another disadvantage of the Naive Bayes classifier which might discourage us from using it is that it works only with categorical variables, so one has to transform continuous features to discrete, which might lead to the loss of a lot of information.

4.2.6 Multivariate Adaptive Regression Splines

We will employ at least one MARS model with an appropriate set-up of parameters as another baseline classifier. The MARS algorithm is also suitable for a large number of predictor variables, robust to outliers, it automatically detects interactions between variables, and despite its complexity, it is an efficient fast algorithm. However, it is susceptible to overfitting, more difficult to understand and interpret as against others, and not good with missing data.

4.2.7 Artificial Neural Network

We will fit at least two different artificial neural networks (ANN) models, specifically with different numbers of layers and different numbers of units. ANN learning algorithms are quite robust o noise in the training data set, they are often used where the fast evaluation of the learned target function is required and its ability to learn and model non-linear and complex relationships. However, the ANN classifier has a disadvantage of unexplained functioning of the network and also no specific rule for determining the structure of artificial neural networks, the appropriate network structure is achieved through experience and trial and error.

4.3 Model Validation

The k-fold cross-validation resampling technique will be used to validate our prediction models.

4.4 Performance/Evaluation Metrics

The following metrics will be employed to examine the performance of the models for best predictive power.

4.4.1 F-beta Score (F_β)

The F-beta score or F-beta measure is the weighted harmonic mean of precision and recall, reaching its optimal value at 1 and its worst value at 0. It turns the F-measure into a configurable single-score metric for evaluating a binary classification model based on the predictions made for the positive class. The beta parameter determines the weight of recall and precision in the combined score. $\beta < 1$ lends more weight to precision, while $\beta > 1$ favors recall. That is, a smaller beta value, such as 0.5, gives more weight to precision and less to recall, whereas a larger beta value, such as 2.0, gives less weight to precision and more weight to recall in the calculation of the score. It is a useful metric to use when both precision and recall are important but slightly more attention is needed on one or

the other, such as when false negatives are more important than false positives, or the reverse (Machine Learning Mastery (2021), Scikit Learn (2021)).

We can compute the F-beta score by

$$F_\beta = \frac{(1+\beta^2) * Precision * Recall}{\beta^2 * Precision + Recall},$$

where **precision** and **recall** are defined below.

The choice of the beta parameter will be used in the name of the F-beta score. For example, a β value of 1 is referred to as the F_1-measure or the F_1 score. A β value of 2 is referred to as F_2-measure or F_2-score.

Precision

Precision is a metric that quantifies the number of correct positive predictions made. It is computed as the ratio of correctly predicted positive classes divided by the total number of positive classes that were predicted as in the formula below.

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{\text{True Positives}}{\text{Total Predicted Positives}}$$

Recall

Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made. The intuition for recall is that it is not concerned with false positives and it minimizes false negatives (Machine Learning Mastery, 2021). It is calculated as

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{True Negatives}} = \frac{TP}{TP+NP}$$

4.4.2 Area Under the Receiver Operator Curve (AUROC)

The AUROC curve is a graphical illustration of the performance of the prediction model. The ROC curve is the relationship between the recall and precision over varying threshold values. The threshold is the positive predictions of the model. The AUROC curve is plotted by keeping the x-axis a false positive rate and the y-axis as a true positive rate. Its value ranges from 0 to 1 (Theerthagiri, Jeena Jacob, Usha Ruby, and Yendapalli, 2021).

4.4.3 Area Under the Precision Recall Curve (AUC-PR)

The area under the precision-recall curve (AUC-PR) is a model performance metric for binary responses that is appropriate for rare events and not dependent on model specificity. Precision-recall curves have also been

recognized as useful for classification performance assessment for unbalanced binary responses in bioinformatics. Like AUC-ROC, AUC-PR is a threshold-independent metric that calculates the area under a curve, where the curve is defined by a trade-off between different aspects of performance as the threshold applied to the model's predictions varies. Both ROC and PR curves are functions of the confusion matrix (Sofaer, Hoeting, and Jarnevich, 2019).

4.5 Model Selection

Base on the calculated metrics for each classifier algorithm, the best model will be selected, for instance, the model with the highest AUROC will be best among the others, however, AUROC alone is not the measure for selection, hence a contingent table will be obtained with classifier models on the row and metric on the column and base on all the metric estimates the one with the best performance across all metrics will be our final predictive model for analysis.

4.6 Model Assessment

Our final obtained classifier model will be validated using the test data provide. That is we will use our model to predict on test data given, specifically looking at some important measures such as specificity and sensitivity of the predicted outcome by the model. Also, we seek to have a prediction accuracy of approximatley 98% and above for our model.

5 Model Deployment

To aid easy deployment and implementation of our proposed model(s), we plan to develop a web application with the R Shiny App package.This dashboard will provide users, among other features, the flexibility to interact with the visualizations and models to assess performance based on changing parameters. We are aiming at an application similar to the online prediction tool developed by Dominguez-Rodriguez et al. (2021).

References

Dominguez-Rodriguez, S., Villaverde, S., Sanz-Santaeufemia, F. J., Grasa, C., Soriano-Arandes, A., Saavedra-Lozano, J., ... others. (2021). A bayesian model to predict covid-19 severity in children. *The Pediatric Infectious Disease Journal*, *40*(8), e287–e293.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Machine Learning Mastery. (2021). A gentle introduction to the fbeta-measure for machine learning. Retrieved October 15, 2021, from https://machinelearningmastery.com/fbeta-measure-for-machine-learning/

Scikit Learn. (2021). Retrieved October 15, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.fbeta_score.html

Sofaer, H. R., Hoeting, J. A., and Jarnevich, C. S. (2019). The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, *10*(4), 565–577.

Theerthagiri, P., Jeena Jacob, I., Usha Ruby, A., and Yendapalli, V. (2021). Prediction of covid-19 possibilities using k-nearest neighbour classification algorithm. *Int J Cur Res Rev| Vol*, *13*(06), 156.

Wikipedia. (2021). Support-vector machine. Retrieved October 15, 2021, from https://en.wikipedia.org/wiki/Support-vector_machine