Predicting Evacuation Location for Post-Disaster Recovery Modeling

Aaron Appelle [*1] **Ana Moura-Cook** [*1] **Davyd Tamrazov** [*1]

Abstract

The field of post-disaster recovery modeling is currently limited by semi-heuristic assumptions. The ability to establish robust relationships is hindered by categorical, survey-based data. This project explores the use of regression methods, decision tree models, and neural networks to predict a household's post-disaster location of temporary shelter, a decision with significant implications on population out-migration. The resulting models perform substantially better than a naive classifier but illustrate the challenges of class-imbalanced data.

1. Introduction

1.1. Motivation

Disasters are becoming increasingly frequent across the United States and worldwide, causing devastating losses in economic downtime, damaged infrastructure, community wellbeing, and more. An affected region can take years to remediate such losses and return to its pre-disaster levels. To aid in post-disaster decision making, researchers in the field of disaster resilience have begun developing comprehensive models of regional recovery. These models aggregate individual households' actions to summarize regional outcomes, but are often limited by the broad use of semi-heuristic assumptions.

This project explores various machine learning techniques to predict an important decision: where a displaced household will seek temporary shelter immediately following a disaster, based on a vast range of demographic and socioeconomic factors. The effects of staying in a hotel, with family and friends, or in a public shelter can influence a household's decision about whether to remain in the area and contribute to the rebuilding process or to move away permanently. Taken at scale, it's the difference between timely recovery and mass regional population out-migration.

[*]Equal contribution [1]Department of Civil and Environmental Engineering, Stanford University, CA, United States.

1.2. Related Works

Much of the data available in this field are survey-based, and therefore primarily categorical with long-tail distributions. Post-disaster recovery has been previously studied following the 2011 Tohoku earthquake to identify significant drivers of business recovery [1], which implemented the Absolute Shrinkage and Selection Operator (LASSO) [2] paired with the Synthetic Oversampling Minority Technique (SMOTE) [3] to address class imbalance. In a similar study, Random Forest Regression was used to identify which characteristics predict tornado preparedness [4]. These models do not yet have widespread use in the industry due to lack of precedent.

As we are most interested in correctly identifying minority cases, we also leverage prior research in the area of machine learning for imbalanced datasets. Recent advances in this area include Focal Loss [5] by Facebook AI Research, which prevents the easy negative examples from overwhelming the training loss by focusing training on a sparse set of hard examples. The most common methods for learning class-imbalanced datasets are introducing weighting terms on the loss functions and re-sampling examples [6, 7, 8, 9, 10, 11]. There are also attempts to develop rigorous theory behind the behavior of common algorithms like SGD Momentum on imbalanced datasets [12]. This is an active research area, but many modifications for deep learning are most reliable with large training set sizes.

2. Dataset and Features

2.1. Overview

The data considered are the public use, anonymized responses to the 2017 American Housing Survey. Administered by the United States Bureau, this survey is the "most comprehensive national housing survey in the United States" [13]. The raw dataset for the larger San Francisco area (including Oakland and Hayward) contains 3,343 variables and 2,286 observations in total. Most of the survey questions produce categorical data, much of it binary. The inputs of the model are described in Table 1.

The output feature predicted is the household's response to the question, "If you had to evacuate from your town or city to a safe place at least 50 miles away for at least two weeks, where would you most likely stay during those two weeks?"

The options include staying (1) with relatives or friends, (2) at a public shelter, (3) at a hotel, (4) in an RV, and (5) other.

2.2. Preprocessing

Out of the original dataset, 995 examples have valid responses associated with the output variable. Of these 995, 44 responses (or 4.4%) indicated they would evacuate to an RV or "other" location, and are thus removed. The remaining 951 examples with output classes (1) through (3) constitute 95.6% of the valid dataset. The data are then divided into training, validation, and test sets with a 60-20-20 split.

We remove certain features from consideration: annotative variables, which indicate whether the data for that variable were modified in post-processing, and any variables missing more than 25% of responses. We then leverage pertinent domain knowledge to infer which questionnaire categories are likely to relate to disaster preparedness and which are useful for the domain of disaster recovery modeling. For example, demographic and financial information for a household is relevant and easy to obtain in the future, making it well-suited for inclusion in the model. The categories of 61 selected features are shown in Table 1.

Category	Description
Occupancy & Tenure	Months occupied, vacancy, seasonal characteristics
Structural	Exterior features, interior features
Demographics	Age, sex, race, disability, number of members, multi-generational
Income	Household income, personal income
Housing Costs	Affordability, utilities, value, debt
Neighborhood	Gated community, nearby features, school quality, social capital, ratings
Recent Movers	Reason for moving, housing search

Table 1. Model Input Features in AHS Dataset

2.3. Principal Components Analysis

A principal components analysis (PCA) helps us better understand which attributes explain evacuation location. The first and second principal components explain 11.4% and 8.7% of the variance respectively, meaning that the remaining 79.9% of variance is explained by lesser components. Output labels are shown plotted against the first two principal components in Figure 1. This analysis demonstrates that the data lie in many dimensions and cannot easily be reduced to fewer. We conclude from the PCA that further selection of features will not significantly improve the models' outcomes, as the data do not exhibit a strong signal in any small number of dimensions.

Following the analysis, the principal component variable loadings are normalized and scaled in order to rank the features' importance. The features that contribute most to the first two principal components are the *number of rooms*, *owner or renter status* of the unit and *monthly total utility amount*. These are compared to the features deemed most important by decision tree models in Section 4.2.

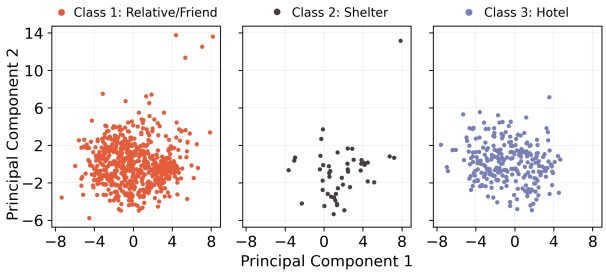

Figure 1. Principal Components Analysis on AHS Dataset

2.4. Encoding

Three methods are considered for encoding the categorical variables: label encoding, one-hot encoding, and entity embedding. Label encoding assigns an integer value to each class within a given variable, but can result in ranking of different options when no such meaning exists in the labels themselves. One-hot encoding solves this problem by portraying each label as a distinct orthogonal vector in k-dimensional space for k classes, but this substantially increases the dimensionality of the dataset. Entity embedding [14] is a blend of the prior methods in which each class is represented by a fixed-length vector. Similar categories are close to each other in the embedding space, helping the neural network to generalize better. Label encoding is found to be the best option for most methods, as the majority of our categorical variables are binary and ranking is not a significant issue. Entity embedding is used in the neural networks to improve their performance.

3. Methods

3.1. Imbalanced Classes

The AHS dataset is imbalanced, with 67% of respondents evacuating to a friend's or relative's home (Class 1), 5% evacuating to a shelter (Class 2), and 28% evacuating to a hotel (Class 3). We account for this imbalance using dataset augmentation via SMOTE [3] which generates new training responses in non-majority classes based on the existing responses [15]. We further mitigate the risk of the train, validation, and test sets having different proportions of non-majority class examples by employing a stratified data splitting technique, which ensures that each subset has the same proportion of class examples. Finally, instead of over-sampling, loss functions such as Focal Loss and

Balanced Cross Entropy are used in the neural networks. These more severely penalize improper labeling of minority classes as described in Section 1.2.

We prioritize predicting deviations from the norm of the majority class by evaluating every method by its balanced accuracy, the average of recall on each class:

$$\tilde{A} = \frac{1}{k}\sum_{j=1}^{k}\frac{T_j}{n_j}$$

where T_j is the number of correctly-classified examples and n_j is the number of observations in class j.

3.2. Ridge Regression

We consider Ridge Regression (RR) as the baseline for comparison with more complex models. RR is a multi-output linear regression method regularized with the L2-norm of the coefficients, in which a separate classifier is trained for each class using a one-vs-all approach [16]. In each binarized model, the target class is encoded as 1, the rest encoded as -1, and the loss function is the L2-norm penalized residual sum of squares. The minimum of the loss function is found analytically using normal equations:

$$\min_{\theta\in\mathbb{R}^2}\frac{1}{2}\|X\theta - y\|_2^2 + \frac{\lambda}{2}\|\theta\|_2^2 \;\rightarrow\; \hat{\theta} = \left(X^\top X + \lambda I\right)^{-1} X^\top y$$

3.3. Decision Tree Methods

3.3.1. Random Forest (RF)

Decision tree methods are a family of algorithms for classification problems which work by sequentially splitting the data at a number of features to minimize Gini impurity defined by the likelihood of an incorrect classification [16]. The random forest method is a bagging-based algorithm that averages many randomly generated, uncorrelated decision trees, thereby generating unbiased predictions.

3.3.2. XGBoost

XGBoost [17] is an efficient implementation of the gradient boosting algorithm that leverages a set of weak decision tree learners to create a more robust classifier. This algorithm is based on the principle of using gradient descent in an iterative fashion to "boost" weak prediction models. This is done by fitting a decision tree to the residuals from the ensemble model defined by the previously fit decision trees. The objective function uses multinomial softmax loss with regularization terms penalizing high tree complexity:

$$\ell(\theta)^{(t)} = \sum_{i=1}^{n}\ell_{\text{CE}}(y^{(i)}, \hat{y}^{(t)}) + \gamma T + \frac{1}{2}\lambda\sum_{i=1}^{T} w_j^2$$

where $\hat{y}^{(t)}$ is the prediction value at step t; λ is the regularization term; and γ, T and w are the parameters that define the complexity of the tree at step t.

3.4. Neural Networks (NN)

We implement neural networks in Pytorch [18] structured to perform multi-class classification on an imbalanced dataset. The model architecture is illustrated in Figure 2.

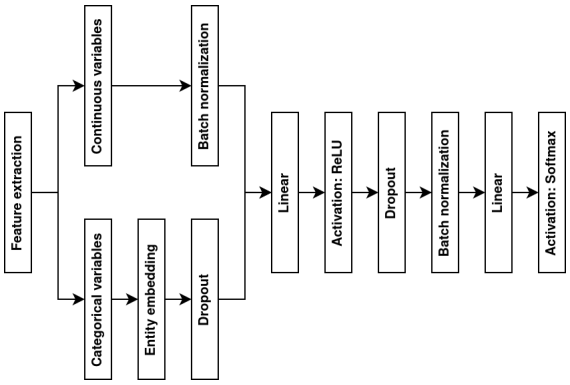

Figure 2. Neural Network Architecture

We use ReLU as the activation function for the hidden layer and incorporate batch normalization and dropout layers. Adam [19] is used as the optimizer during training, which performs first-order gradient-based optimization of stochastic objective functions and is commonly used for neural networks. The details of the parameters are described in Section 4.3. In the next Sections 3.4.1 and 3.4.2, we describe the different loss functions for neural networks.

3.4.1. Weighted Cross-Entropy Loss (NNWCE)

Cross entropy loss measures the divergence between the predicted and actual labels, which is equivalent to using the negative log likelihood loss after applying the softmax activation function. NNWCE re-weights the loss function to alter the thresholds for predicting each class, such that the model is biased towards predicting the minority classes:

$$\ell(x^{(i)}) = w_{y^{(i)}}\left(-S^{(i)}_{y^{(i)}} + \log\sum_{j=1}^{k}\exp(S^{(i)}_j)\right)$$

Where $w_j = (1 - n_j/n)$ is the weight for class j (constant for all examples), and n_j is the true class count of class $j \in \{1, \ldots, k\}$ over the dataset of length n.

3.4.2. Focal Loss (NNFL)

Focal loss also emphasizes minority class accuracy, but it further discriminates by which examples are "easy" to classify and which are "hard;" that is, those examples that are near the decision boundary between two or more classes. The easily-classified examples contribute less to the loss than the hard-to-classify examples, thereby prioritizing accuracy on the borderline cases:

$$\ell(x^{(i)}) = -\alpha(1 - p^{(i)}_{y^{(i)}})^{\gamma}(\log(p^{(i)}_{y^{(i)}}))$$

where $p^{(i)} = [p_1^{(i)}, \ldots, p_{k-1}^{(i)}] \in [0,1]^k$, γ is the focusing parameter that increases loss due to misclassified predictions and α is the weighting factor to magnify minority classes.

4. Results and Discussion

The results from the five methods are summarized in this section. The metrics of interest (Table 3) include total and balanced accuracies, the weighted F1 score, and the area under the receiver operating characteristic (ROC) curve for each class. Of these, balanced accuracy is considered the most important. Each method's ROC curve and confusion matrix are illustrated in Figure 3. In sum, our models perform substantially better on balanced accuracy than the trivial classifier, but the tradeoff between total and balanced accuracy is somewhat limited by the data.

4.1. Ridge Regression Results

RR performs poorly in the classification of minority classes, with the recall for Class 2 and Class 3 being 0.45 and 0.43, respectively. This is expected for our highly imbalanced dataset. While RR does include a regularization term to decrease the variance in the fitted parameters, nothing in the formulation biases results towards minority classes. RR has the potential to be an interpretable model by analysis of coefficients. However, we find that the coefficients reveal the weaknesses of the performance, as it computes opposing relationships between two very similar variables:

Variable	Class 1	Class 2	Class 3
Household Income	$4e{-}07$	$-2e{-}07$	$-2e{-}07$
Family Income	$-6e{-}07$	$1e{-}07$	$5e{-}07$

Table 2. Coefficients on two variables in RR model

Given this contradiction and others, we acknowledge that the model's interpretability is limited by its accuracy.

4.2. Decision Tree Results

The Random Forest model is tuned to maximize the balanced accuracy of the validation set while keeping the decision tree ensemble as simple as possible to produce a robust classifier. As such, tree depth is limited to 3 with 100 estimators. To further prevent overfitting, the minimum number of nodes at each leaf is set to 3. In a similar fashion, XGBoost hyper-parameters are set to the maximum tree depth of 4 with 500 estimators and the optimal convergence was found to be with the learning rate 0.1. To avoid overfitting, $L2$ regularization with a strength of $5e2$ is applied to the loss function, while column and instance subsamples are set to 0.1 and 0.3 respectively to fit a tree to a small subset of observations and features.

The decision tree methods identify the most salient features for predicting the outcome by computing the mean decrease impurity resulting by splitting at each feature over all ensembles. These differ substantially between the two methods, as seen in Figure 4, though there are three common features: *insurance per month*, *household income*, and the *year the respondent arrived in the United States*.

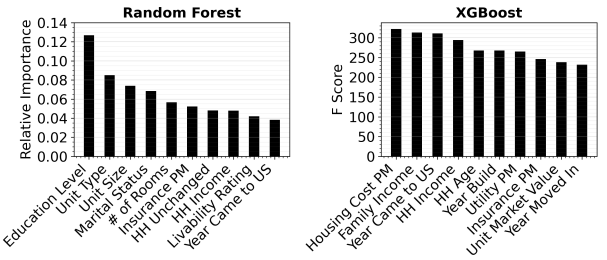

Figure 4. Feature importance scores using decision tree models

Despite the differences, both models identify features that are directly or indirectly related to wealth, education level, and how long the household has lived in the region, which corroborates findings of previous related studies [20, 21, 4]. We posit that wealthy, well-educated, and settled residents are more likely to stay at a hotel or with family and friends than in a public shelter in the post-disaster context.

The XGBoost and RF models have similar performance, both achieving higher balanced accuracy than the Ridge Regression model. This is primarily due to decision tree models better capturing the non-linear boundary between classes, which enables more accurate prediction of Class 2 as corroborated by the higher AUROC scores (Table 3).

4.3. Neural Network Results

NNWCE is implemented with weights of $w_1 = 0.33, w_2 = 0.95$, and $w_3 = 0.72$ per the equation based on class sizes in Section 3.4.1. The number of nodes in the hidden layer (104) is selected as $2/3$ of the input nodes plus the number of output nodes. The dropout rate is set to 0.1 and 0.5 respectively for input and hidden nodes to increase the robustness of the fit while preserving the integrity of the input layer. Since the hidden layer is large, we run the algorithm for 200 epochs with a learning rate of $1e{-}4$ and an aggressive weight decay of $1e{-}2$ applied to avoid overfitting. We chose a batch size of 100, as a larger batch size means that the gradient is computed based on a larger number of examples, increasing the chances of choosing the correct descent direction and decreasing sensitivity to variations in the batch makeup. Note that no over-sampling is performed for the NN models.

NNFL is parameterized by α and γ, set to 0.05 and 3.0 respectively, which significantly amplify the mean loss due

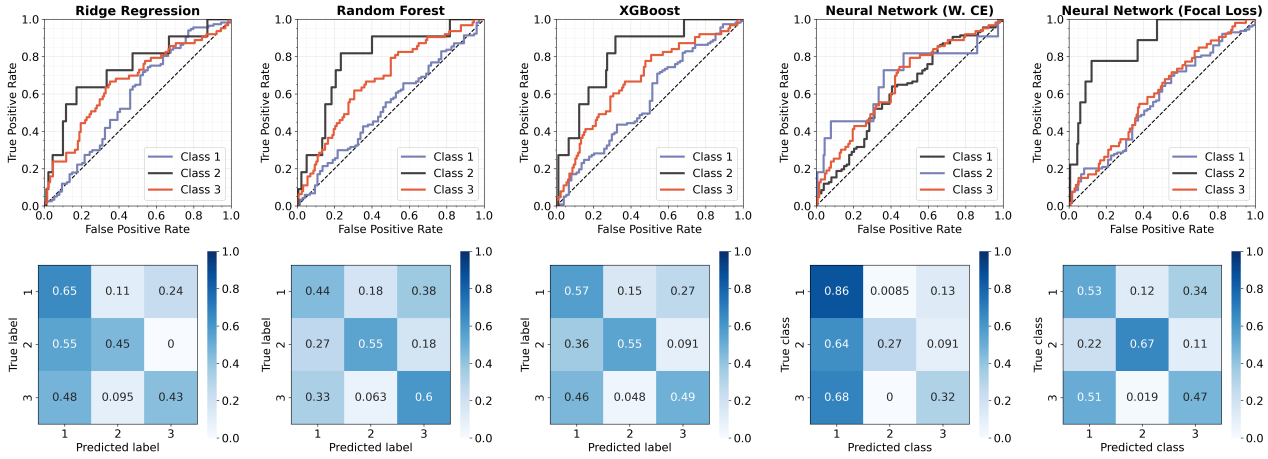

Figure 3. Receiver operating characteristic curves and normalized confusion matrices

Metric	Baseline *Naive*	Ridge Regression	Decision Tree *Random Forest*	Decision Tree *XGBoost*	Neural Network *Weighted CE*	Neural Network *Focal Loss*
Total Accuracy	0.6126	0.5654	0.5026	0.5445	**0.6492**	0.5864
Balanced Accuracy	0.3333	0.5109	0.5310	0.5367	0.4845	**0.5739**
Weighted F1 Score	0.4654	0.5740	0.5171	0.5575	**0.6180**	0.5976
AUROC (Class 1)	0.5000	0.5835	0.5337	0.5698	**0.6185**	0.5678
AUROC (Class 2)	0.5000	0.7354	0.7864	0.8131	0.6843	**0.8651**
AUROC (Class 3)	0.5000	0.6612	**0.6773**	0.5820	0.6598	0.5956

Table 3. Summary of Test Set Results

to misclassified minority class samples. In this case, we found that the algorithm converges to higher validation set accuracy with the hidden layer size set to 50 nodes. The algorithm is run for 1200 epochs with a smaller learning rate of $1e{-}5$ and a weaker weight decay of $1e{-}4$.

We find that NNFL performs best, reaching 57% balanced accuracy and 59% total accuracy. NNWCE has slightly higher total accuracies, but cannot obtain above 50% balanced accuracy. Notably, the final NNFL model outperforms all of the other classifiers in predicting Class 2, thereby achieving the highest balanced accuracy. While the neural networks lack a degree of interpretability, they depend on far fewer assumptions about household decision-making than state-of-the-art models for predicting disaster-preparedness. Nevertheless, neural networks were unable to significantly improve upon the deterministic tree-based models as it is more difficult to generalize tabular data with probablistic methods.

5. Conclusion

We do not identify a single best choice model; rather, the neural networks and XGBoost might be appropriate for different purposes. NNFL achieves the best balanced accuracy and is preferred when we are interested in predicting the number of households which will evacuate to shelters. The reason is that Focal Loss explicitly emphasizes minority class accuracy per its formulation. However, only a small proportion of households (approx. 5% of the dataset) will evacuate to shelters. When we are more interested in evacuations to hotels, NNWCE has a higher AUROC performance for Class 3 (on par with the best Random Forest model) while still maintaining high total accuracy. XGBoost has the benefit of identifying salient features and performs relatively well across error metrics and across classes, but it is not best-in-class for any one metric.

Future work on this topic can implement novel classification techniques for imbalanced and long-tailed datasets, including the methods from Section 1.2. LDAM and deferred re-weighting until after the initial stage [7] are good candidates for this dataset. Data undersampling and better class weighting schemes can also be explored in the future. Establishing additional data-driven relationships will substantially improve existing recovery models' semi-heuristic assumptions. Models can be developed to predict outcomes ranging from federal aid applications, to hazard insurance coverage, to homeowners' decisions to reconstruct their property, all of which contribute to the larger picture of regional recovery.

Contributions

Aaron, Ana, and Davyd contributed equally and worked collaboratively on all parts of the project. The team implemented supervised and deep learning methods in Python together using Git to share the code and worked simultaneously on the report. Rodrigo Costa advised the project topic selection and initial model development. Aakanksha N.S. [22] provided the baseline framework for entity embedding and the neural networks. All of our code is available on Github: https://github.com/amouracook/229project.

References

[1] R. Costa and J. Baker, "Factors influencing business recovery after the 2011 tohoku earthquake," 2020.

[2] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178

[3] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 321–357, 06 2002.

[4] J. Choi, S. Robinson, R. Maulik, and W. Wehde, "What matters the most? understanding individual tornado preparedness using machine learning," *Natural Hazards*, vol. 103, pp. 1183–1200, 2020.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[6] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," 2019.

[7] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," 2019.

[8] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Deep imbalanced learning for face recognition and attribute prediction," 2019.

[9] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[10] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, p. 249–259, Oct 2018. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2018.07.011

[11] Z. Liu, P. Wei, J. Jiang, W. Cao, J. Bian, and Y. Chang, "Mesa: Boost ensemble imbalanced learning with meta-sampler," 2020.

[12] K. Tang, J. Huang, and H. Zhang, "Long-tailed classification by keeping the good and removing the bad momentum causal effect," 2020.

[13] "American housing survey national public use file," United States Census Bureau, 2017. [Online]. Available: https://www.census.gov/programs-surveys/ahs/data/2017/ahs-2017-public-use-file--puf-/ahs-2017-national-public-use-file--puf-.html

[14] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," 2016.

[15] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[17] T. Chen and C. Guestrin, "Xgboost," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[20] A. Fothergill and L. A. Peek, "Poverty and disasters in the united states: a review of recent sociological findings," *Natural Hazards*, vol. 32, pp. 89–110, 2004.

[21] E. Ablah, K. Konda, and C. L. Kelley, "Factors predicting individual emergency preparedness: a multi-state analysis of 2006 brfss data," *Biosecur Bioterror*, vol. 32, pp. 317–330, 2009.

[22] A. N.S., "Deep learning for tabular data using pytorch," 2020. [Online]. Available: https://towardsdatascience.com/deep-learning-for-tabular-data-using-pytorch-1807f2858320