DATA 1030 Final Report: Predicting Community Problem Solvers in CA

Alejandro Contreras, Ph.D. Student in Political Science at Brown University
alejandro_contreras@brown.edu

GitHub Repository: https://github.com/acontreras001/Data_1030_Final

Introduction:

Why study community problem solvers?
Helping to address issues in one's local community is vital for a properly functioning society. Unfortunately, it can be difficult to address local issues for a variety of reasons. This raises the question of who is most likely to act and attempt to solve community issues? Put differently, is it possible to predict using machine learning algorithms who will most likely attempt to resolve community problems? This question is important as if one is able to claim that certain characteristics are more likely to result in community problem solvers, one could in theory attempt to increase the number of community problem solvers through those characteristics.

What data is being used?
With the question in mind, this project attempts to produce an answer using an annual survey from UCLA's School of Public Health, the California Health Interview Survey (CHIS). Specifically, I rely on data from the 2020 iteration of the survey, which focuses on adults, and contains a total of 21,949 respondents and 604 features. The data are gathered with race/ethnicity, language, and geographic location in-mind to create a representative sample of California residents. Additionally, the data are gathered using both web and phone surveys.

As mentioned, the dataset contains a total of 604 features. These features can be divided into 13 sections, and I utilize variables from Section A (Demographics Part I), Section G (Demographics Part II), Section K (Employment, Income, Poverty Status, Food Security), Section M (Housing and Social Cohesion), and Section P (Voter Engagement). My target variable (coded as "am3") is a categorical feature on community problem solving. In particular, respondents are asked the question, "In the past 12 months, have you volunteered to organize or lead efforts to help solve problems in your community?". Respondents can respond, "Yes" or "No" which results in this project being a straight forward classification problem.

What previous work has used this data?
Finally, the annual survey has been used in the past for policy briefs as well as peer-reviewed published articles. This includes work on issues like disability services (Kietzman et al. 2022), hate against Asian Americans, Native Hawaiians, & Pacific Islanders (Shimkhada & Ponce 2022), and food insecurity (Lowery et al. 2022). More published work that utilized the California Health Interview Survey can be found here: https://healthpolicy.ucla.edu/publications/latest/Pages/default.aspx.

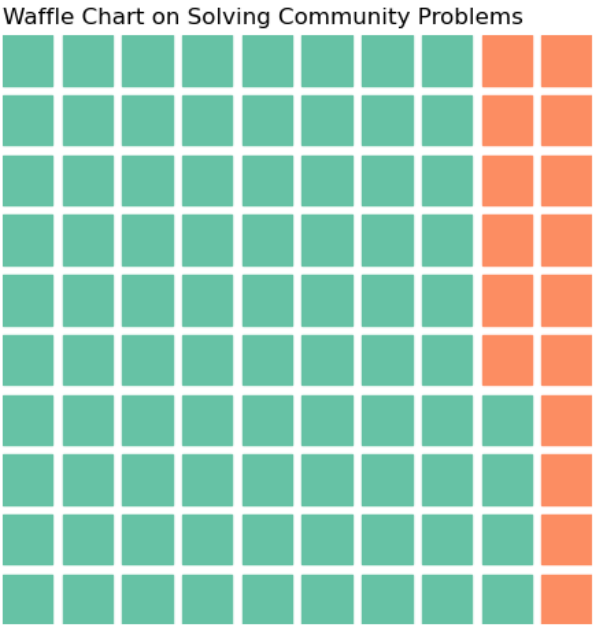
Exploratory Data Analysis (EDA):

As mentioned in the previous section, there are over 604 variables in the dataset. Given constraints, I chose to focus on 12 variables (11 predictor variables and 1 target variable) from the CHIS dataset. Specifically, I chose the variables based on demographic information, economic information, and past political engagement. A full list of the variables can be found in the table below:

Variable code	Variable Description	Variable Type	Variable Purpose
am39	Community Problem Solver	Categorical	Target
ak7_p1v2	Length of time employed at main job	Categorical	Predictor
ak3_p1v2	Number of hours worked per week	Categorical	Predictor
srsex	Sex/Gender	Categorical	Predictor
citizen2	Citizenship	Categorical	Predictor
ag22	Military	Categorical	Predictor
ur_bg6	Urban/Rural	Catergorical	Predictor
ak25	Housing	Catergorical	Predictor
famsize2_p1	Family Size	Continuous	Predictor
srage_p1	Age	Ordinal	Predictor
sreduc	Education Level	Ordinal	Predictor
ap73v2	Voting Frequency in Presidential Election	Ordinal	Predictor

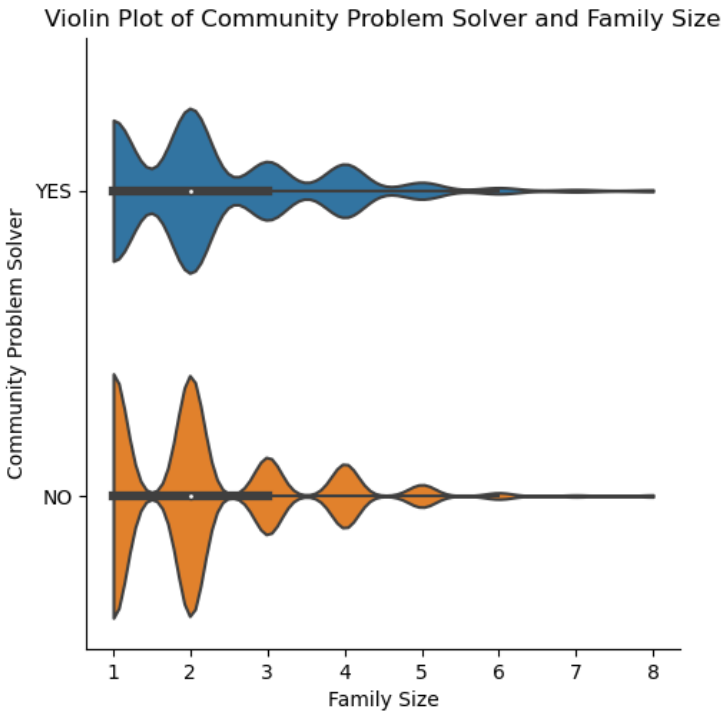
I chose these variables as they comprised of five sections of the survey that I think are pertinent to my question. With this information, I created multiple figures, and in this report, I share four figures.

The first figure is a waffle chart that shows the distribution of the target variable. As demonstrated, the vast majority of respondents (84.4%) answered "No" to the question, "In the past 12 months, have you volunteered to organize or lead efforts to help solve problems in your community?" Consequently, 84.4% is the baseline accuracy that any machine learning algorithm should have. This is due to the fact that the algorithm could guess "No" for every respondent and be correct 84.4% of the time. Additionally, it is important to note that this should be considered an imbalanced dataset, as the ideal scenario would be to have 50% of the responses classified as "No" and 50% of responses classified as "Yes."



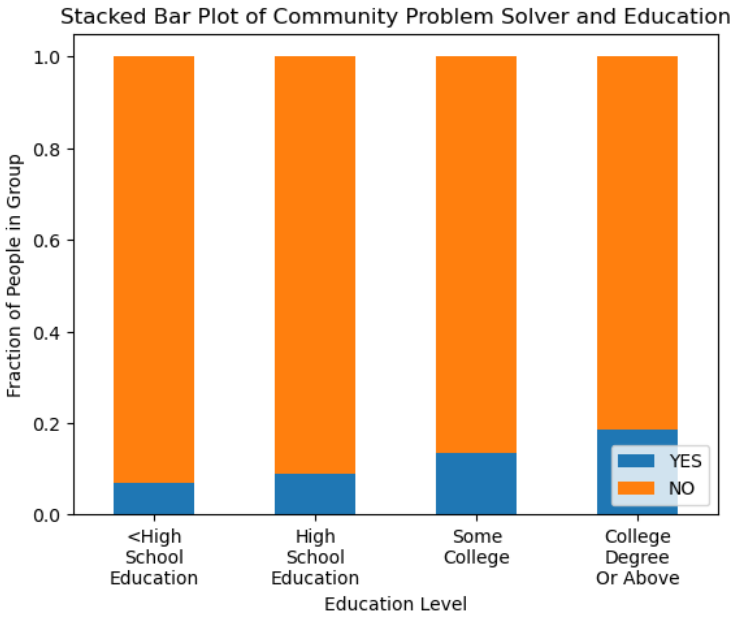
Caption: This is a waffle chart that demonstrates that 84.4% of respondents answered "No" to the question "In the past 12 months, have you volunteered to organize or lead efforts to help solve problems in your community?" Conversely, 15.6% of respondents answered "Yes."

The second figure is a violin plot that graphs the respondent's family size on the x-axis and whether the respondent is a community problem solver on the y-axis. As one can see, the size of the peaks and valleys is slightly larger for the "No" responses although the graphs generally contain a similar shape.

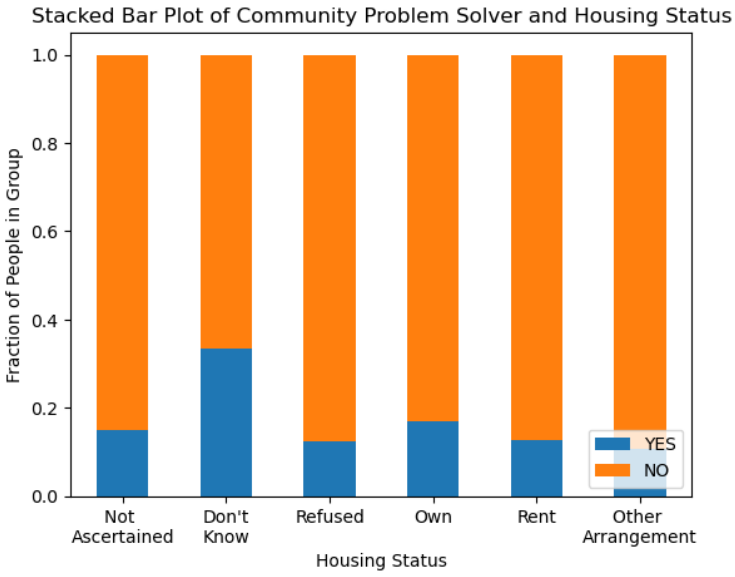


Caption: This is a violin chart that demonstrates that plots community problem solvers against family size. As demonstrated by the peaks and valleys, there are some slight differences between the figures although, overall they could be considered quiet similar.

The third figure is a category-specific bar chart that plots one's education level against the target variable. As demonstrated, as one moves across the x-axis, the fraction of "Yes" responses increases. This signifies that education level may be positively correlated with the target variable.



Finally, the fourth figure is a category-specific bar chart that plots the target variable against the respondent's homeownership status. As one can see, there is some variation based on the homeownership status, although it is difficult to tell whether these differences are significant in any way.



Methods:

Splitting, Preprocessing, and ML Pipeline

The splitting strategy that I chose to implement was the basic **train_test_split** command from the **SKLearn environment**. The strategy is appropriate for survey data given that it is independently and identically

distributed, i.e., the responses of one survey participant should not impact the responses of another survey participant. Specifically, I used a 60-20-20 split so that 60% of the observations were in the train set, 20% of the observations were in the test set, and 20 % of the observations were in the validation set.

After splitting the data, I created a pipeline using the **ColumnTransformer**. Specifically, I used the **OneHotEncoder** to convert the categorical features to dummy arrays, the **OrdinalEncoder** to convert ordinal features to integer arrays, and the **MinMaxScaler** to transform the continuous features. Afterward, I applied the pipeline the train, test, and validation datasets.

ML Algorithms and Hyperparameters

After preprocessing the data, I used Logisitic Regression (no penalty, l1 penalty, l2 penalty, and elesticnet penalty), Random Forest Classification, KNearestNeighbor Classification, and SVM Classification to answer my question. For each of these algorithms, I tuned two hyperparameters. The hyperparameters for each algorithm can be seen in the following table:

ML Algorithm	1st Hyperparameter	2nd Hyperparamter
Logistic Regression	alpha = np.logspace(-2,2,21)	rho = np.linspace(0,1,num = 21)
Random Forest	max_features = [1, 3, 10, 30, 100]	max_depth = [0.25, 0.5,0.75,1.0]
KNearest Neighbor	n_neighbors = [1, 10, 30, 100]	weights = ['uniform','distance']
SVM Classification	C = [1e-3, 1e-1, 1e0, 1e1, 1e3]	gamma = [1e-3, 1e-1, 1e0, 1e1, 1e3]

Evaluation Metric

With regard to an evaluation metric, the typical choice for evaluating unbalanced datasets are either the f_betascore or the precision-recall curve. However, as will be described in more detail in the following section, neither evaluation is useful given the results.

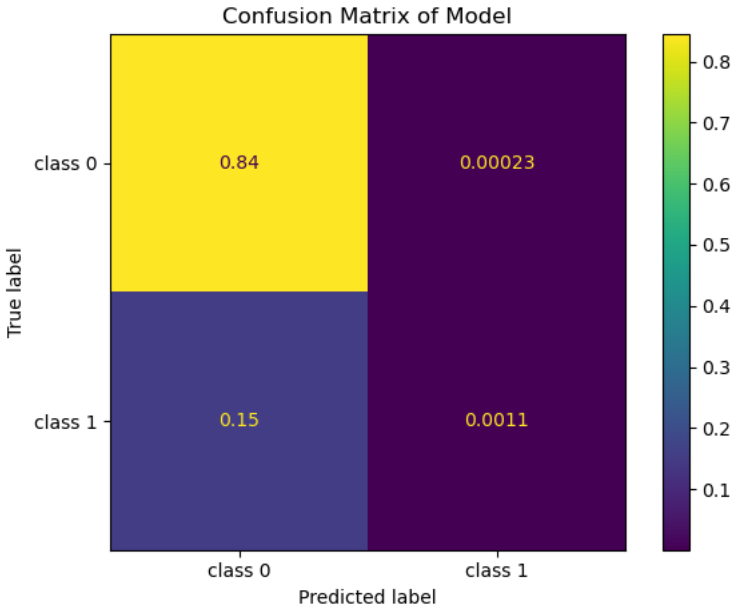
Results:

Comparison of Model Performance

Unfortunately, all of the machine learning algorithms produced null results (they did not perform better than the baseline model). This is clearly demonstrated in the following table, which documents the accuracy score for each machine-learning algorithm. Note the hyperparameter combination is not listed on the table, as multiple hyperparameter combinations resulted in the same accuracy score.

ML Algorithm	Accuracy
Baseline Model	84.43%
Logistic Regression	84.44%
Random Forest Classification	84.56%
KNearestNeighbor	84.46%
SVM Classification	84.46%

Essentially, regardless of the model and hyperparameter combination, nearly all of the algorithms chose to predict 0 ("No") for nearly all of the observations. Below is an example of a confusion matrix from one of the RandomForest models.



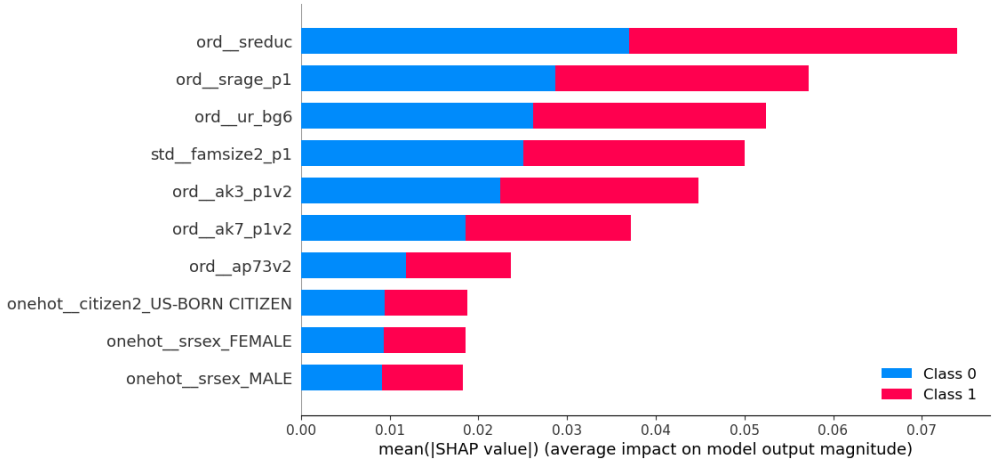
With this in mind, it is important to remember that for imbalanced datasets, it is typically recommended to not use accuracy and the ROC curve as evaluation metrics since they rely on the True Negatives in the Confusion Matrix. Instead one should normally rely on the f_beta score (normally with a higher beta value) or the precision-recall curve. However, in my scenario, neither is a good metric since none of the models performed better than the baseline model. For the f_beta score, the value was 0 regardless of what beta value was used. This makes sense theoretically, as both Precision and Recall scores rely on the number of True Positives. Given that the number of true positives is 0 (or nearly 0), one will always receive an f_beta score of 0 regardless of what beta value is used. Similarly, regardless of what algorithm and hyperparameter combination is used, one should receive a precision-recall curve of about 15%. This is because the evaluation metric relies on the number of positive cases divided by the total number of cases.

Feature Importance

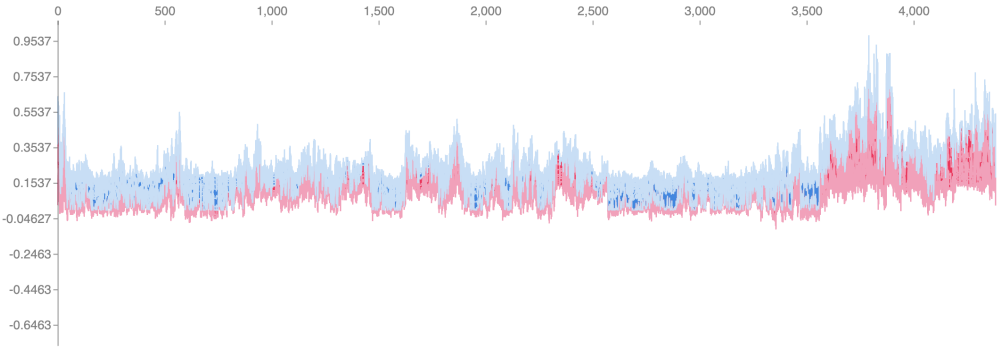
Preface: Given the results, the feature importance of each model should not be trusted. Regarless, I describe the feature importance given that it is a requirement.

If one were to utilize the **permutation_importance** feature from the **SKLearn environment**, one would see that the top three features for the Logistic Regression, KNearestNeighbor, and SVM Models were "ord__ak7_p1v2" (Length of time working at main job), "ord__ak3_p1v2" (# of hours worked per week), "onehot__ak25_RENT" (Renter Status). Alternatively, in the Random Forest model, the features that were the most important were "std__famsize2_p1" (Family Size), "ord__ak3_p1v2" (# of hours worked per week), "ord__ak7_p1v2" (Length of time working at main job).

When looking at global SHAP values for the Random Forest algorithm, one notices education, age, and urban/rural environment are the features that contain the most importance. A full list of the top 10 most important features for the Random Forest algorithm can be found in the following figure:



When one looks at the local SHAP values, one notices that the values vary depending on the index i.e., the features contain positive or negative effect varies depending on the row. This is demonstrated in the following figure where the x-axis is the index number and the y-axis shows the effect size of the features. Ultimately, the figure serves to show how the same features may contain different weights depending on the index.



Again, given the null results of all of the models, one should not place any stock in the feature importance.

Outlook:

Provided more time, there are a number of additional improvements that could be made to this project. First and foremost, the use of additional features could help to create models which improve the accuracy, f_betascore, and precision-recall curve. This is possible given that I only looked at a sample of the original dataset, and there are still about 500+ additional features that I could potentially utilize. Some of the potentially useful data include additional demographic information, as well as data on other forms of community involvement (ability to contact a public official, voting in a local election, etc.).

Apart from using additional data, some additional improvements that a future iteration of this project would include using additional techniques like XGBoost as well as tuning additional hyperparameters. These improvements, however, come secondary to using additional features, and they would also require a more advanced computer system.

References from Published Works

Kietzman KG, Haile M, Chen X, & Pourat N. 2022. *Demand for Aging and Disability Services Is Increasing in California: Can We Meet the Need?* Los Angeles, CA: UCLA Center for Health Policy Research.

Lowery, B. C., Swayne, M., Castro, I., & Embury, J. 2022. *Mapping EBT Store Closures During the COVID-19 Pandemic in a Low-Income, Food-Insecure Community in San Diego.* Preventing chronic disease, 19, E37. https://doi.org/10.5888/pcd19.210410

Riti Shimkhada and Ninez A. Ponce, 2022: *Surveying Hate and Its Effects During the COVID-19 Pandemic Among Asian Americans and Native Hawaiians and Pacific Islanders* American Journal of Public Health 112, 1446_1453, https://doi.org/10.2105/AJPH.2022.306977