Predicting Community Rates of Homelessness Using Historical Data

Noah Rousell
Brown University
Github[6]

1. Introduction

Homelessness is a major problem in the United States, with over 582,462 homeless individuals counted in January of 2022[1]. Furthermore, homelessness has increased by 6% since 2017[1]. For these reasons it is important to know which communities are going to suffer from the highest rates in the coming years, especially because of the existence of effective interventions like the Housing First model[2,3]. I hope to develop a model that can predict rates of homelessness at the community level using historical data.

1.1 Dataset

I use a dataset compiled by a team associated with the Department of Housing and Urban Development for the study *Market Predictors of Homelessness*[4].

The dataset has a total of 338 features containing outcome, economic, demographic, housing, safety net, and geographic data. Sources of the data and examples are given in table 1. As the study was conducted several years ago, the years included in the data are 2010 through 2017.

Type	Examples	Source(s)
Outcome	Sheltered/Unsheltered Homeless Counts, Homeless Veterans Counts	HUD Point in Time (PIT) Counts
Economic	Median household income, income inequality	Census ACS 5-Year Estimates
Demographic	Total male population, total population ages 65+, total white population.	Census Intercensal Population Estimates, Census ACS 5-Year Estimates
Housing	median contract rent for renter-occupied housing units, share of renter-occupied housing units	Census ACS 5-Year Estimates
Safety Net	Total year round shelter beds, share of households with public assistance income to all households	HUD Housing Inventory Counts (HIC)
Geographic	State, census region, community	N/A

Local Policy	count of begging laws, count of sleeping, camping, lying/sitting, and vehicle restriction laws	National Law Center on Homelessness and Poverty (NLCHP)

Table 1. Table of the categories of features and sources contained within the dataset

Interventions to help the homeless are organized into communities known as Continuums of Care (CoC). Each row in the dataset corresponds to one CoC for a specific year. There are 376 Continuums of Care represented in the dataset spread across all 50 states. With 8 years of data for each community, there are 3008 samples in the dataset.

2. Exploratory Data Analysis

We perform EDA to better understand the distributions of features and their correlations with the target variable. We first visualize the distribution of homeless counts across different communities.

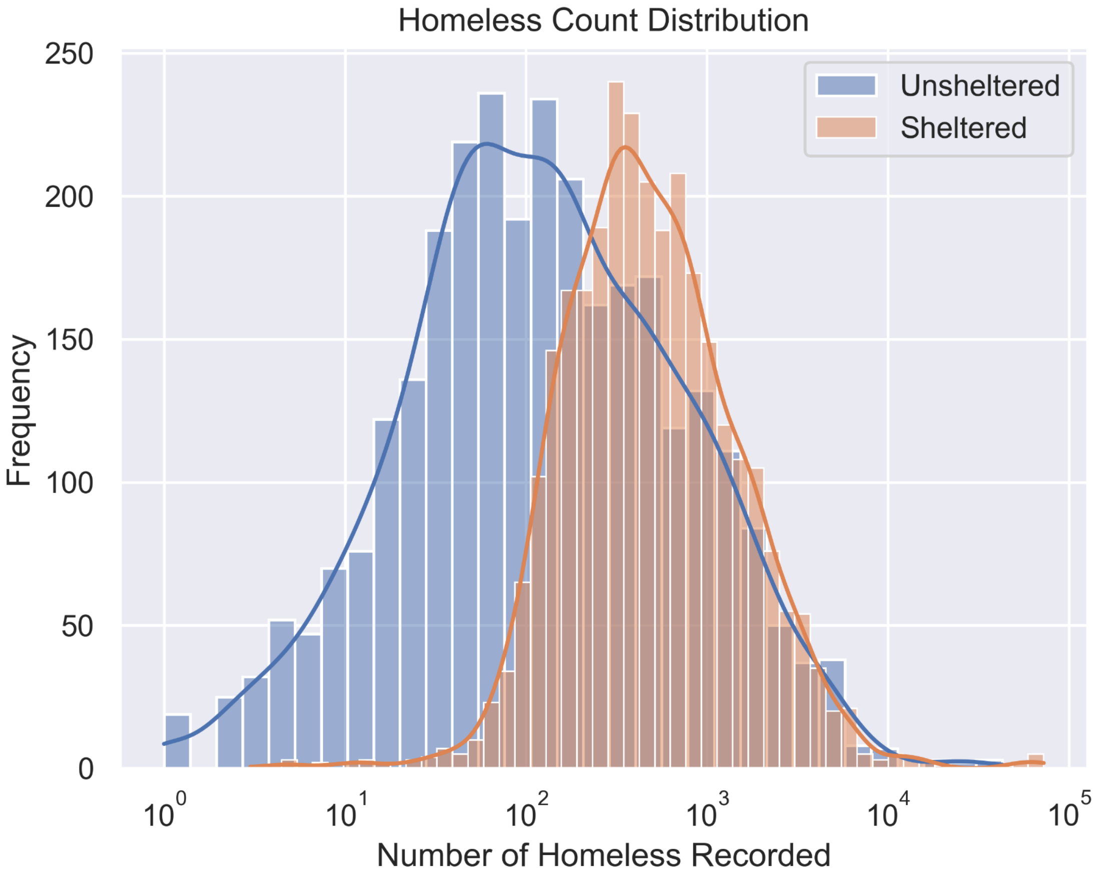

Fig 1. Sheltered and Unsheltered Homeless Counts Distributions. A log scale is used as the data is spread out.

We can see that homeless counts are quite heterogeneous and that the number of sheltered homeless tends to be higher than the number of unsheltered homeless. Some communities are much larger than others population-wise and hence it is more useful to use rates. From now on, we will use individuals homeless per 10,000 as our target variable.

Next let's examine how the rates change over time. In Figure 2, we see rates decreased overall over the 2010-2017 period. Whether a community is rural, suburban, or urban affects rates. Urban and Suburban communities tend to have higher rates of homelessness, not just higher absolute counts.

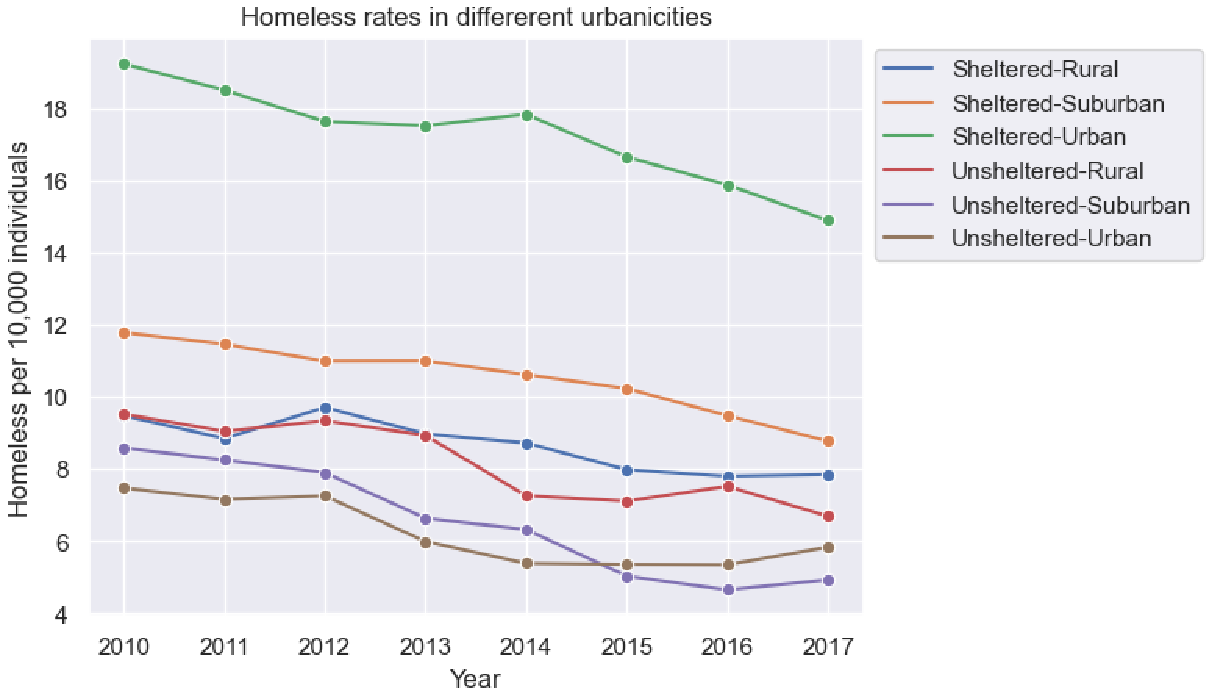

Fig 2. Average homeless rate over time by Sheltered/Unsheltered Homelessness and Urbanicity

Another useful exploration is seeing how rates vary geographically. Figure 3 illustrates that certain states, notably California, Alaska, Oregon, Hawaii, and Florida, have unusually high rates of homelessness. The District of Columbia has the highest overall state rate by far, with a mean rate of 112.5 homeless individuals per 10,000.

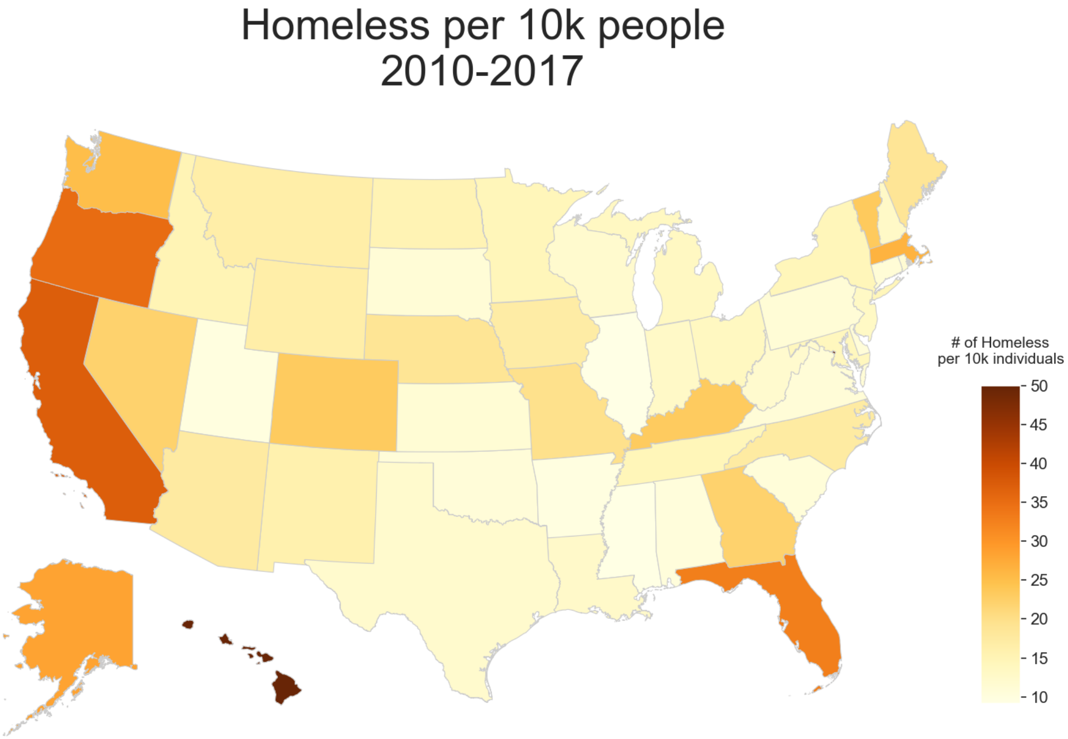

Fig 3. Heatmap of homeless rates by states

We now turn our attention to housing and economic variables correlated strongly with the overall homelessness rate. Due to the veritable mountain of variables in the dataset, we only visualize the variables that have the greatest correlation with the target variable. One such variable is the 4-year change in the Housing Price Index (HPI), with a correlation of 0.42. The HPI broadly measures how expensive single-family homes are. Figure 4 shows that communities that have seen greater rises in housing prices tend to have greater rates of homelessness.

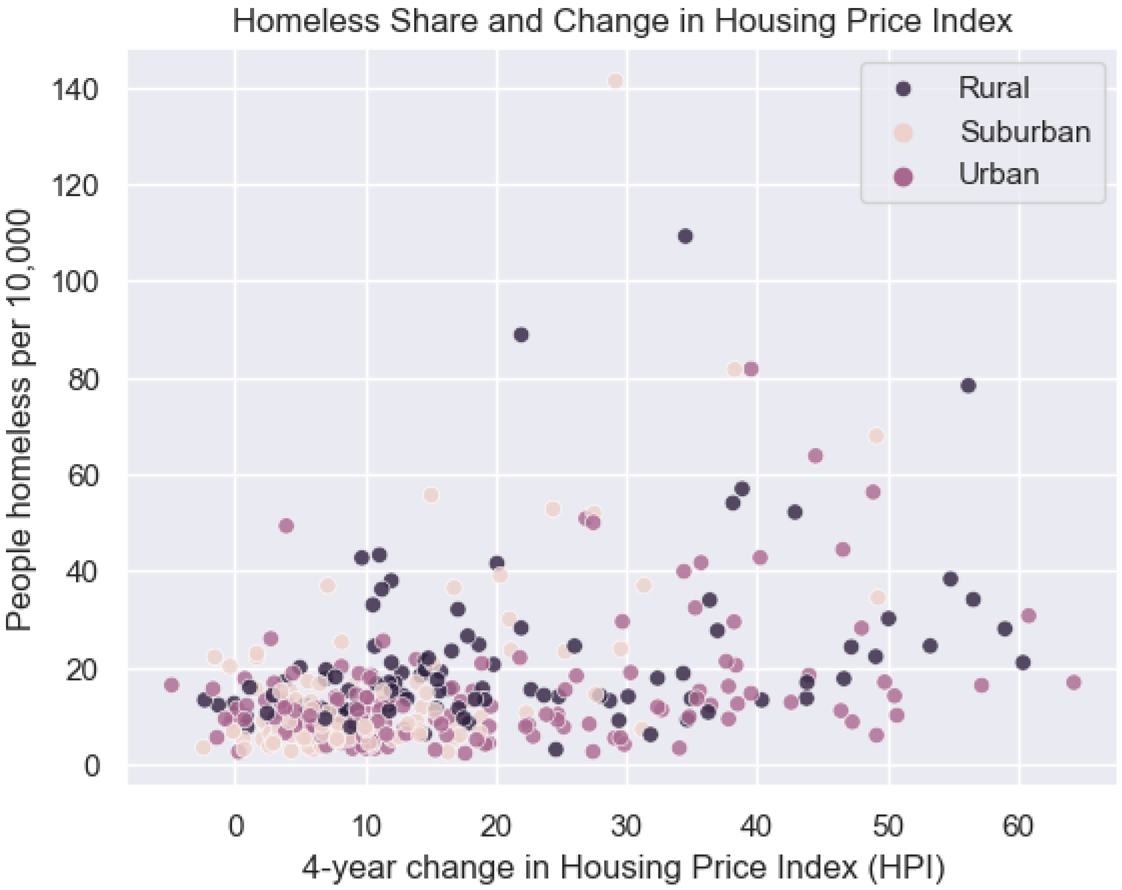

Fig 4. Scatterplot between the 4-year change in the Housing Price Index and Homeless rate. Community urbanicity is denoted by dot color.

The gini index is the classic way to measure inequality within a population, with values closer to 0 representing more egalitarian communities and values closer to 1 more unequal communities. Figure 5 shows the association between between-household inequality as measured by the gini index and community homeless rate. With a correlation of 0.25, this is also a factor the model may weigh more in its calculations. Furthermore, we can see an interaction with urbanicity: higher income inequality leads to a greater homelessness rate especially when it’s paired with rural urbanicity.

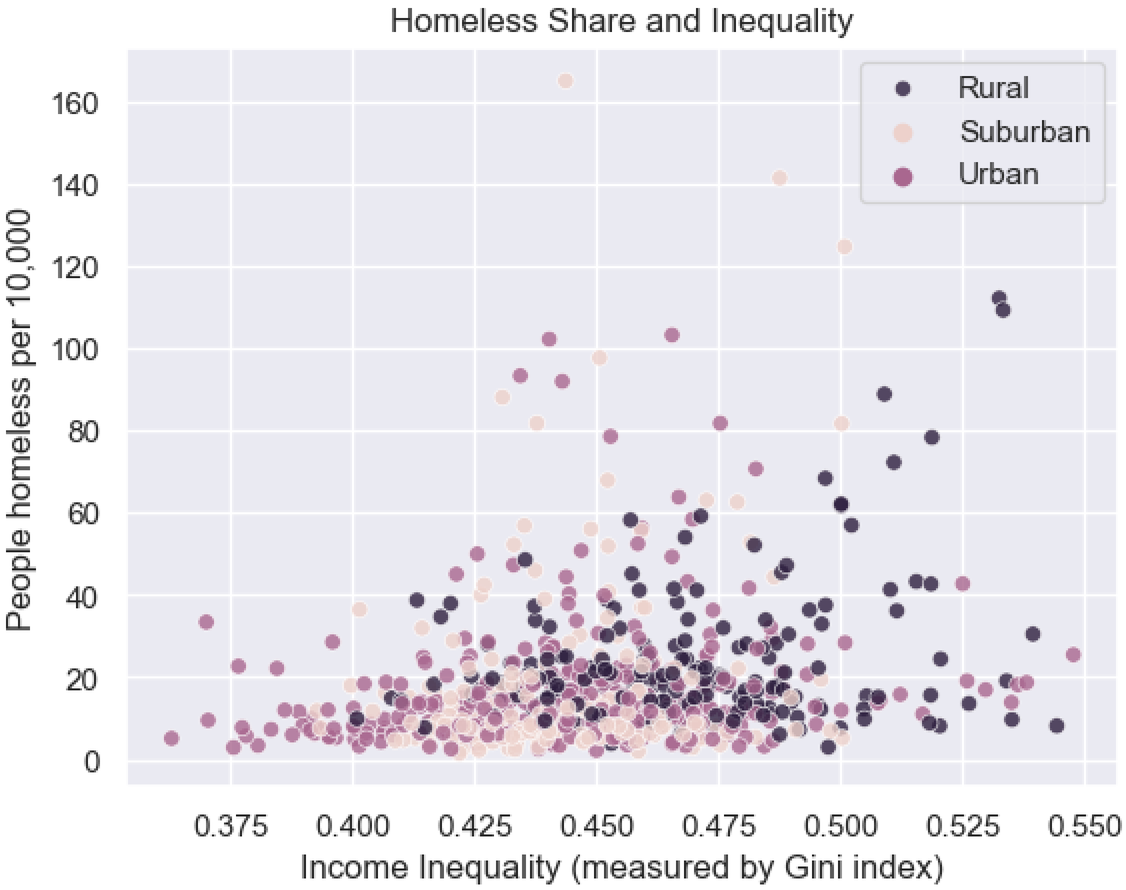

Fig 5. Scatterplot of income inequality between households (as measured by the Gini index) and homeless rates of communities. Community urbanicity is denoted by dot color.

3. Methodology

Model development consists of autoregression, splitting it into train, test, and validation sets, and preprocessing the data.

3.1 Autoregression

With a time-series dataset, features need to be prepared so that they reflect the data that will be available when the model would be used. When predicting homelessness rates for the next year, that year's economic, housing, and demographic data will not be available; we will have to use historical data, letting the models predict using the data from past years. For example, if we want to predict the homeless rate in NYC in 2015, we will provide data from *n* years before (if *n*=3, we will provide data from 2014, 2013, and 2012). We cannot know how many years back it is best to provide (which *n* value), so we will try multiple values of *n*. After doing this, the number of features grows considerably. With *n*=3, we will have thrice the number of features we had before.

3.2 Splitting

We want to train the model with as much data as possible, but it's imperative that we hold out one set of data to tune the models' hyperparameters (validation set) and another to test the model's performance on previously-unseen data (test set).

With a time-series dataset, the value of data across years is not independent, and thus we need to make sure we do not give our models an unfair advantage. This means the data from the train set must come from years strictly before those in the validation set, which must come strictly before years in the test set. Lining up the data by year on the x-axis, Figure 6 visualizes our split.

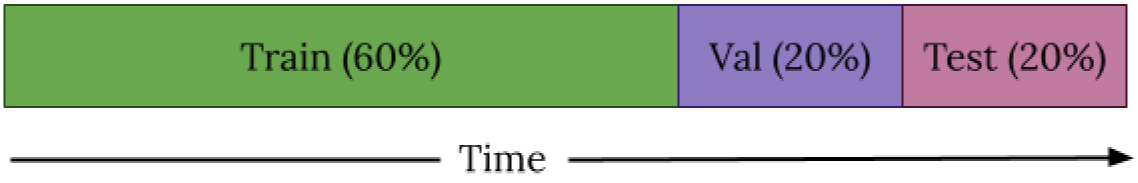

Fig 6. Time-series train/validation/test split illustration

3.3 Preprocessing

Some forms of data, such as text, cannot be processed as-is by ML algorithms. Furthermore, the different scales numerical data can take can obscure feature importances during the interpretation phase. Preprocessing solves these problems.

Most of the dataset's features are numerical yet several, such as *state*, are categorical. Sklearn's OneHotEncoder is used first to deal with these categorical variables, and then StandardScaler is applied universally to ensure linear model weights can be reliably used to interpret feature importances.

3.4 Hyperparameter Tuning and Cross-Validation

We perform hyperparameter-tuning to identify the hyperparameters to use that will reduce overfitting and generalize well to the validation and test sets. For each hyperparameter combination, a separate model is trained and evaluated using the validation set, then the best is selected and evaluated on the test set. For this problem, the number of lagged years of historical data *n* we create becomes another hyperparameter.

We use Root Mean Squared Error (RMSE) as our evaluation metric because it keeps the error in the same units as the target variable.

We train XGBoost on the dataset as-is due to its ability to handle missing values. Unfortunately, the reduced features model and multivariate imputing did not work with this dataset due to the large amount of features with missing data. To utilize other ML algorithms, we create a limited dataset consisting only of the features without missing data. We justify this with the fact that XGBoost feature importance metrics give a low importance to the dropped features.

Models trained and the corresponding hyperparameters tuned are shown in Table 2.

Model	Hyperparameters
XGBoost	reg_alpha (logspace), reg_lambda (logspace).
Lasso/Ridge	reg_alpha (logspace)
Elastic Net	reg_alpha (logspace), l1_ratio (linear)
KNN	n_neighbors (logspace), weights (“distance”, “uniform”)
Random Forest	max_depth (logspace), max_features (linear)
SVR	gamma (logspace), C (logspace)

Table 2. Models trained and hyperparameters used

4. Results

Model evaluation consists of comparing the test scores of models to each other and baselines, interpreting feature importances globally, and examining local feature importances to determine how the model makes predictions for particular points.

4.1 Model Scores

Model scores with *n*=1,3 are visualized in Figures 7 and 8 respectively. Two baselines were created: predicting the target variable’s mean in the training set and predicting last year’s rate.

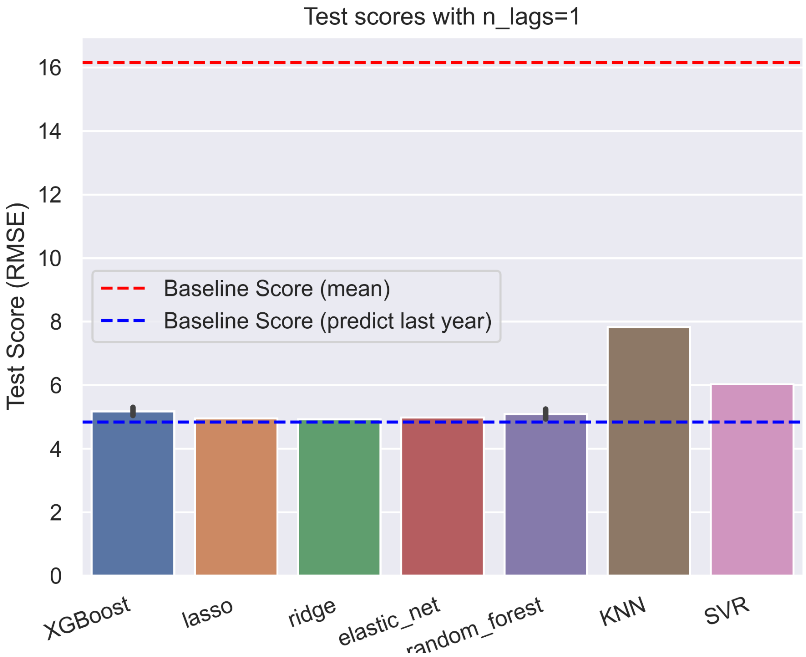

Fig 7. Model test scores compared to baseline with *n*=1 years of historical data. Multiple instances of XGBoost and RandomForest were trained to measure their randomness.

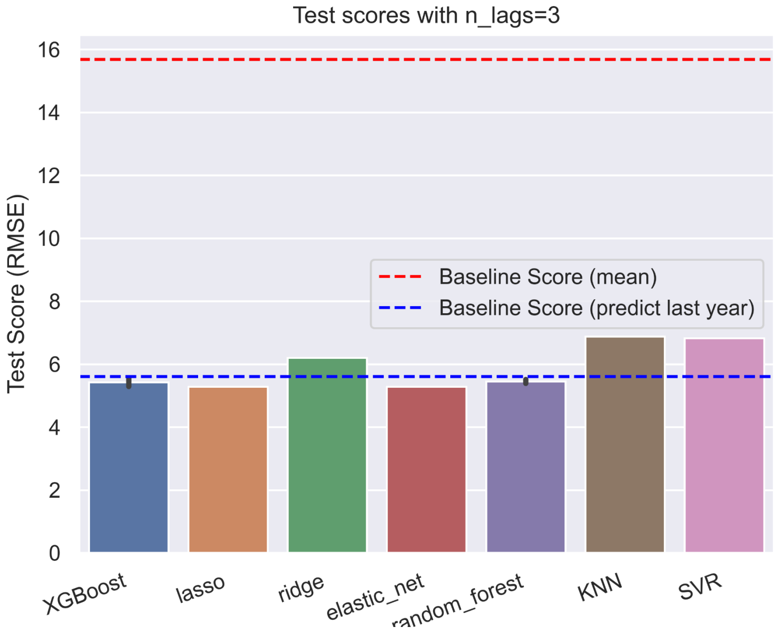

Fig 8. Model test scores compared to baseline with *n*=3 years of historical data. Multiple instances of XGBoost and RandomForest were trained to measure their randomness.

Some models do marginally better than the baseline of predicting last year when *n*=3.

4.2 Interpretation

Interpretation involves finding which features are most important to a model overall (global importance) and which specific feature values are most impactful in making specific predictions (local importance).

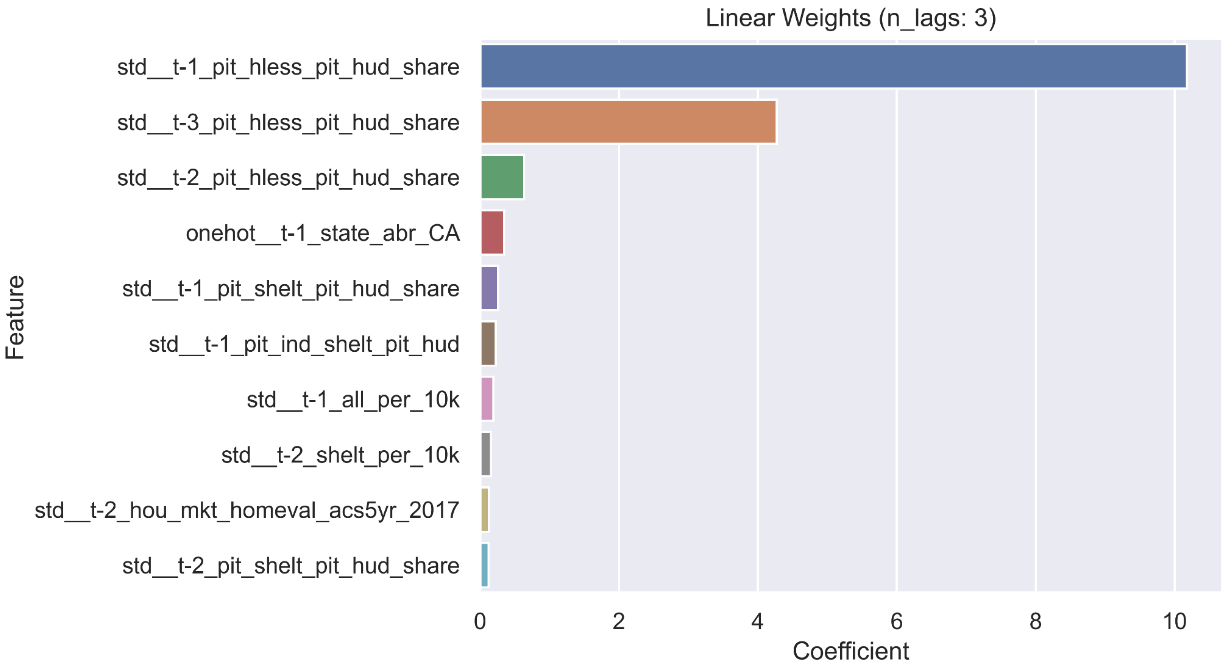

Fig 9. Top 10 Elastic Net Coefficients with *n*=3 years of historical data.

The coefficients of each feature of a linear regression model is a simple way to determine feature importances. Figure 9 shows the top ten weights of the Elastic Net model when *n*=3, which achieved one of the best scores on the test set. We see that the top three greatest weights are given to past rates, suggesting the model starts with last year's rate and follows the trend using the years before.

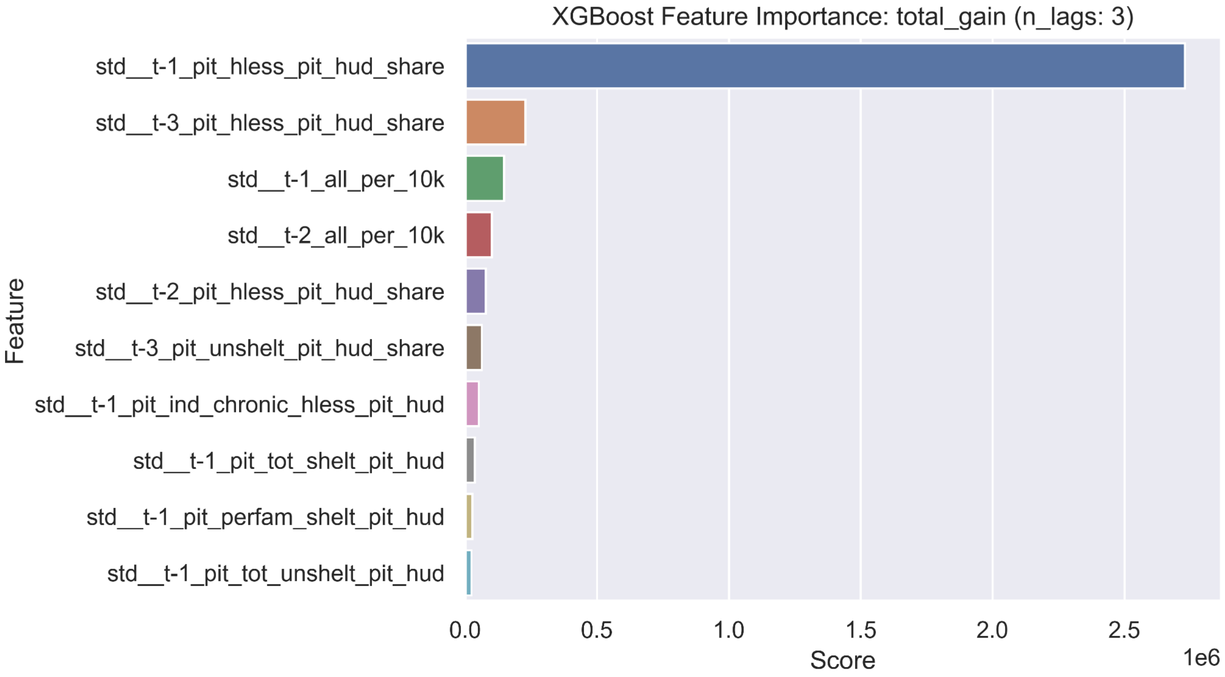

Fig 10. XGBoost total gain feature importance with *n*=3 years of historical data

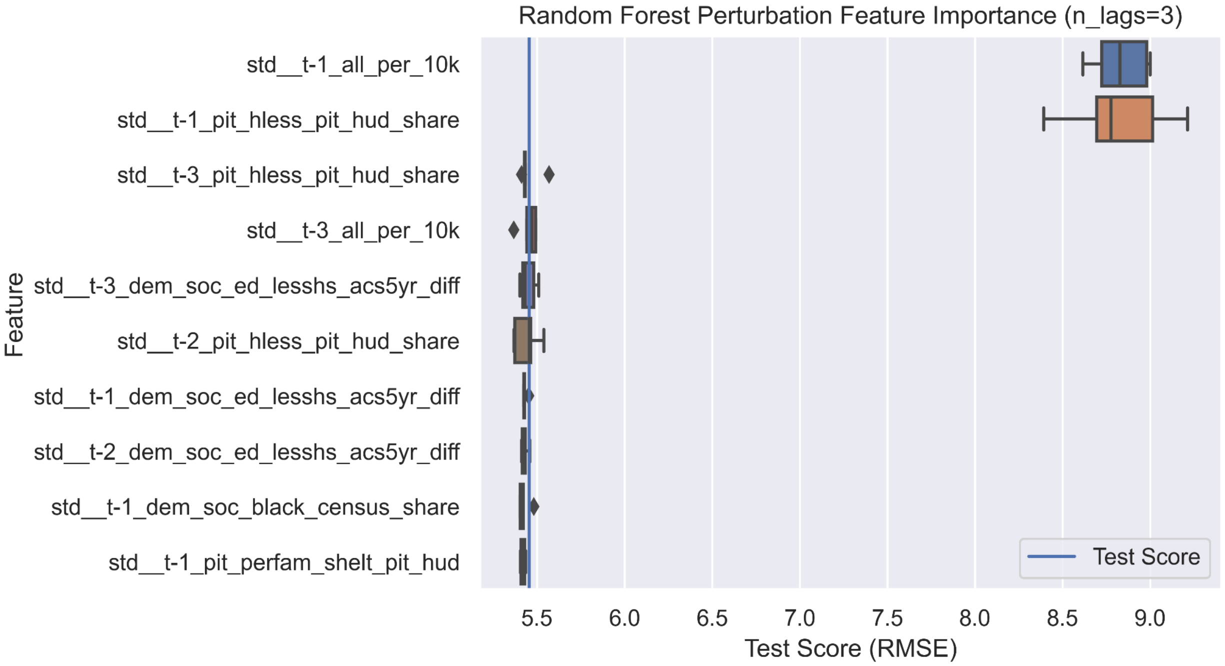

Fig 11. Random Forest Permutation Feature Importance with *n*=3 years of historical data

Figures 10 and 11 tell a similar story, with the most important features for XGBoost (as measured by total gain) and RandomForest (as measured by permutation), being the rates of the previous years.

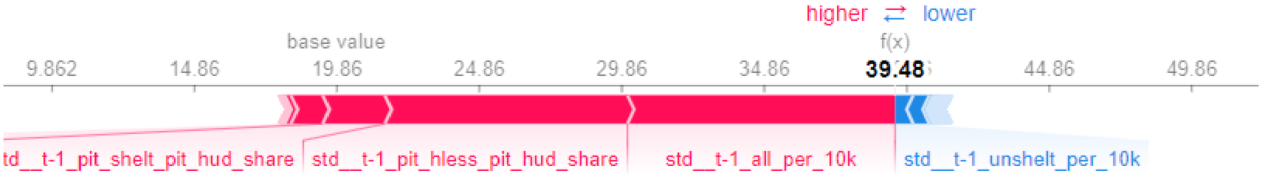

Fig 12. SHAP force plot for random forest model with *n*=1 years of historical data, point 1.

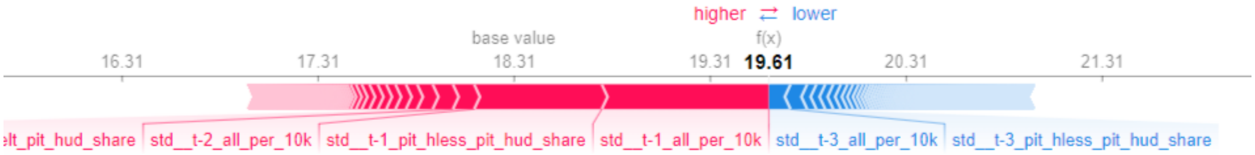

Fig 13. SHAP force plot for random forest model with *n*=3 years of historical data, point 170

Finally, we can examine which features are important for the predictions of specific points using the SHAP library[5], which utilizes game theory to allocate feature importance. Looking at Figures 12 and 13, we notice again that previous years' rates are the most important features.

5. Outlook

While a start, the model created is not significantly better than the baseline of predicting the previous year. There is currently not enough data for the model to learn complex patterns between variables other than past rates.

One thing that could be done is to use folds in the splitting to tune the model with validation sets from more years. Each fold's training set would encompass the last one, with the validation sets falling directly after.

Another thing that could be done is to create a new dataset with data stretching back farther to 2000 and up to the present. This would be made easier by the fact that many of the original features did not end up being particularly predictive, and thus do not need to be included in the expanded dataset.

Finally, setup functions were created so as to make it easy to try out different target variables. It would be interesting to see the performance of models predicting absolute rates of homelessness or specifically unsheltered or sheltered homelessness.

6. References

[1] State of Homelessness 2023

[2] Is the Housing First Model Effective? Different Evidence for Different Outcomes

[3] Effects of Housing First approaches on health and well-being of adults who are homeless or at risk of homelessness

[4] Market Predictors of Homelessness Study

[5] SHAP library

[6] Github Repository: https://github.com/nwrousell/homelessness-prediction