Forecasting Crime and Identifying Key Predictors:

An Interpretable Machine Learning Approach

Han Huang

Data Science Initiative, Brown University

1. Introduction

Purpose

Understanding the factors that influence community-level crime has long been a central objective in sociology, criminology, and public policy. Accurately predicting crime rates can help local governments allocate resources, design prevention strategies, and evaluate the broader social and economic conditions associated with violence and public safety. With the increasing availability of large, multi-source datasets, machine learning provides a powerful framework for studying how structural community features relate to crime outcomes.

Communities and Crime Unnormalized dataset[1]

The Communities and Crime Unnormalized dataset from the UCI Machine Learning Repository, integrates demographic information from the 1990 U.S. Census, policing characteristics from the 1990 Law Enforcement Management and Administrative Statistics (LEMAS) survey, and crime data from the 1995 FBI Uniform Crime Reports. The dataset contains 2,215 U.S. communities and 125 predictive features, covering a wide range of family structure, socio-economic, housing, population composition, and law-enforcement variables. The prediction task focuses on ViolentCrimesPerPop, a continuous variable representing the total number of violent crimes per 100,000 residents, which serves as the dataset's designated goal attribute.

Current Research[2]

This dataset has been widely used in machine learning research, particularly for regression modelings. Prior work shows that predictive performance is generally moderate but limited, reflecting the inherent complexity of social systems. Classical studies using linear regression and shallow decision trees typically report R^2 values around 0.40–0.55, while more advanced models such as Random Forests and Gradient Boosting Machines often achieve R^2 in the range of 0.60–0.75. Importantly, no published work has demonstrated extremely high predictive accuracy, reinforcing that crime is influenced by many unobserved or difficult-to-measure factors.

2.Explanatory Data Analysis

As an initial step of the exploratory data analysis, the extent of missing data across features was examined. Figure 1 summarizes the percentage of missing values for all variables with incomplete observations, with most policing-related features exhibiting high missingness (≈84%).

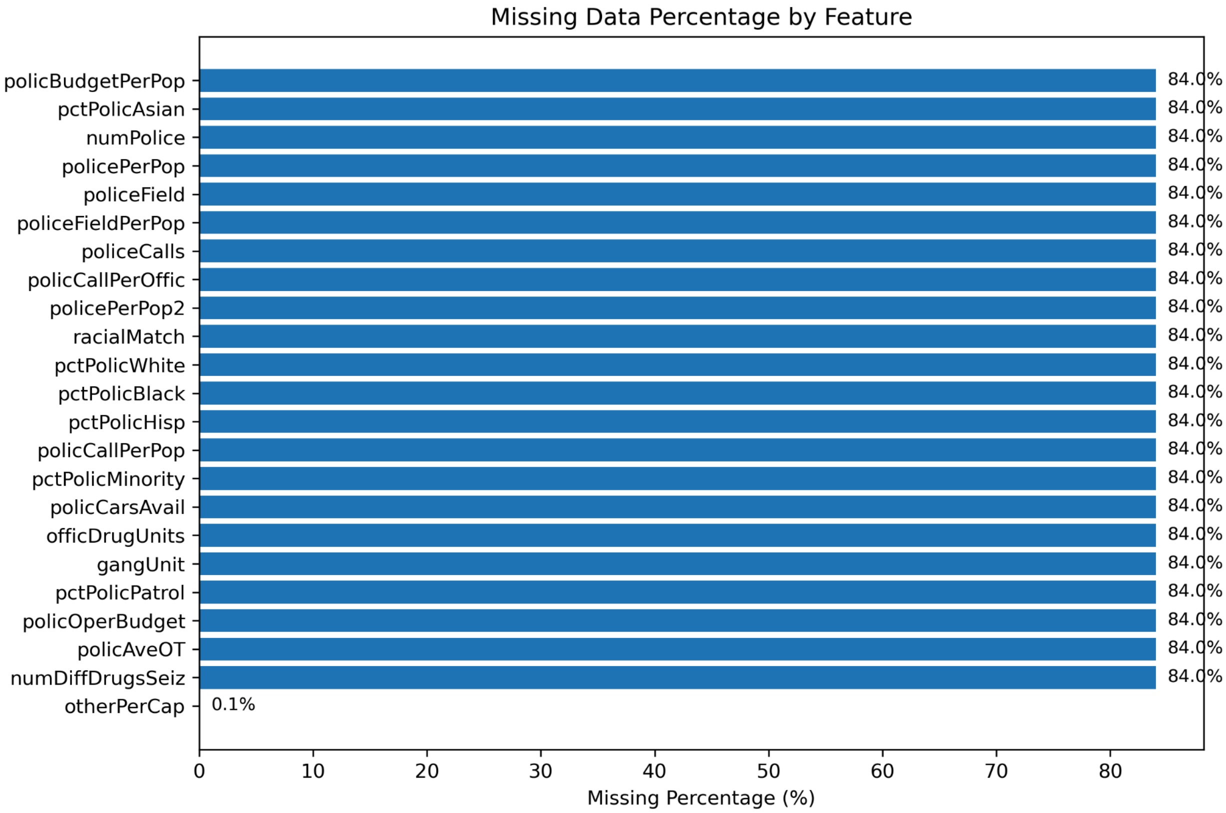

Figure 1. Missing data percentage by feature.

To further explore the relationship between individual predictors and the target variable, bivariate visualizations were used to examine how selected features relate to **ViolentCrimesPerPop**. Figure 2 presents the relationship between **pctKids2Par** and violent crime rates, providing a direct visual assessment of the strength and direction of their association.Higher proportions of children living in two-parent households are associated with lower violent crime rates.

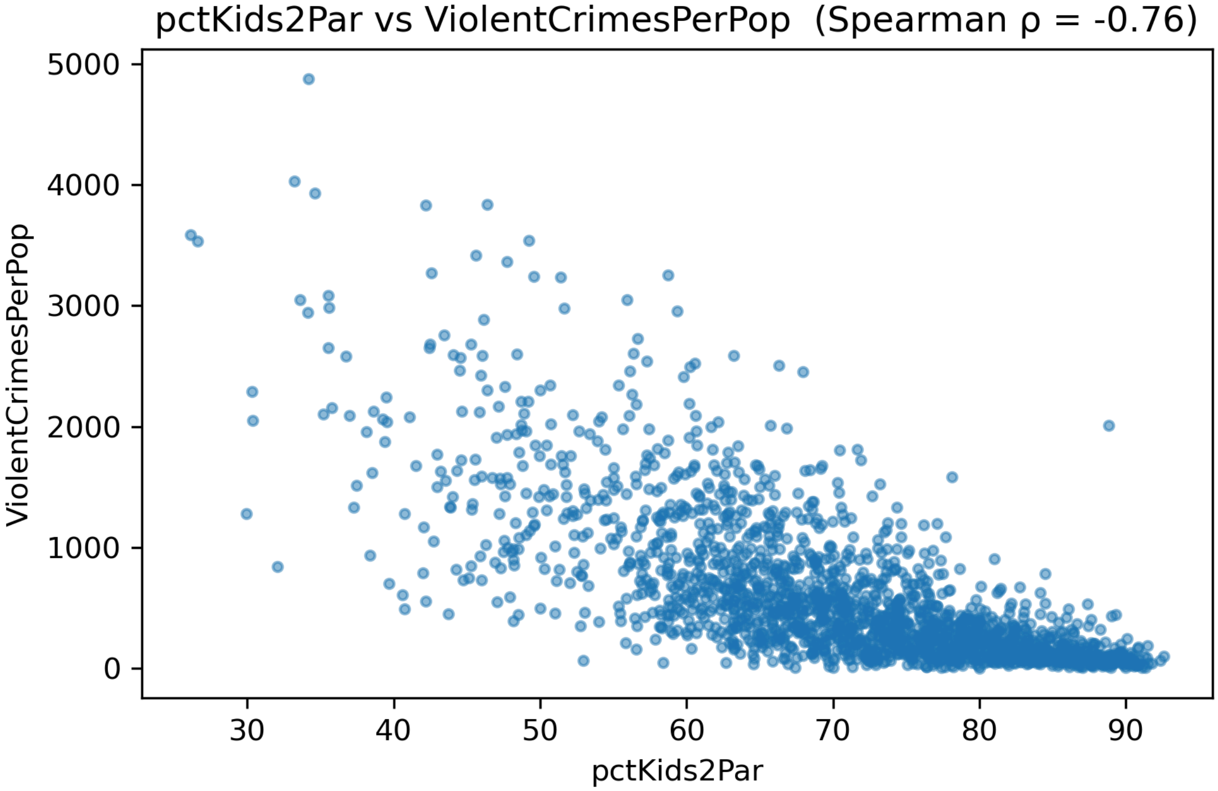

Figure 2. The scatter plot: Relationship between pctKids2Par and ViolentCrimesPerPop.

The relationship between **pctWhite** and **ViolentCrimesPerPop** was also explored to further understand how community demographic composition relates to violent crime rates. Figure 3 visualizes the joint distribution of these two variables using a 2D binned heatmap and reflects a negative association between pctWhite and violent crime rates.

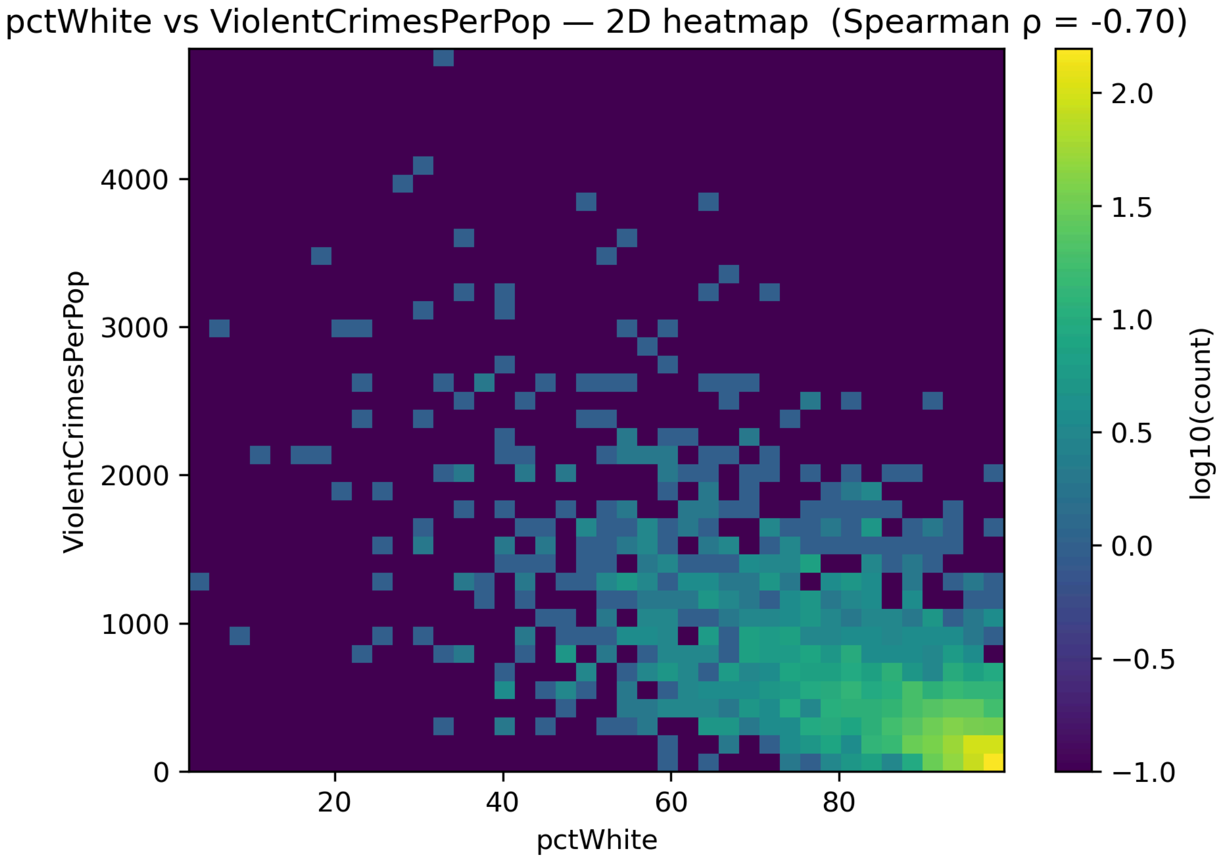

Figure 3. pctWhite vs ViolentCrimesPerPop (2D heatmap).Cells represent the log-scaled count of observations per bin.

Beyond feature-level relationships, the distribution of **ViolentCrimesPerPop** was also examined across U.S. regions to assess potential geographic variation. Figure X presents the distribution of violent crime rates by region using violin plots.Violin plots show differences in the distribution of violent crime rates across regions, with higher central tendencies and greater variability observed in the South and West.

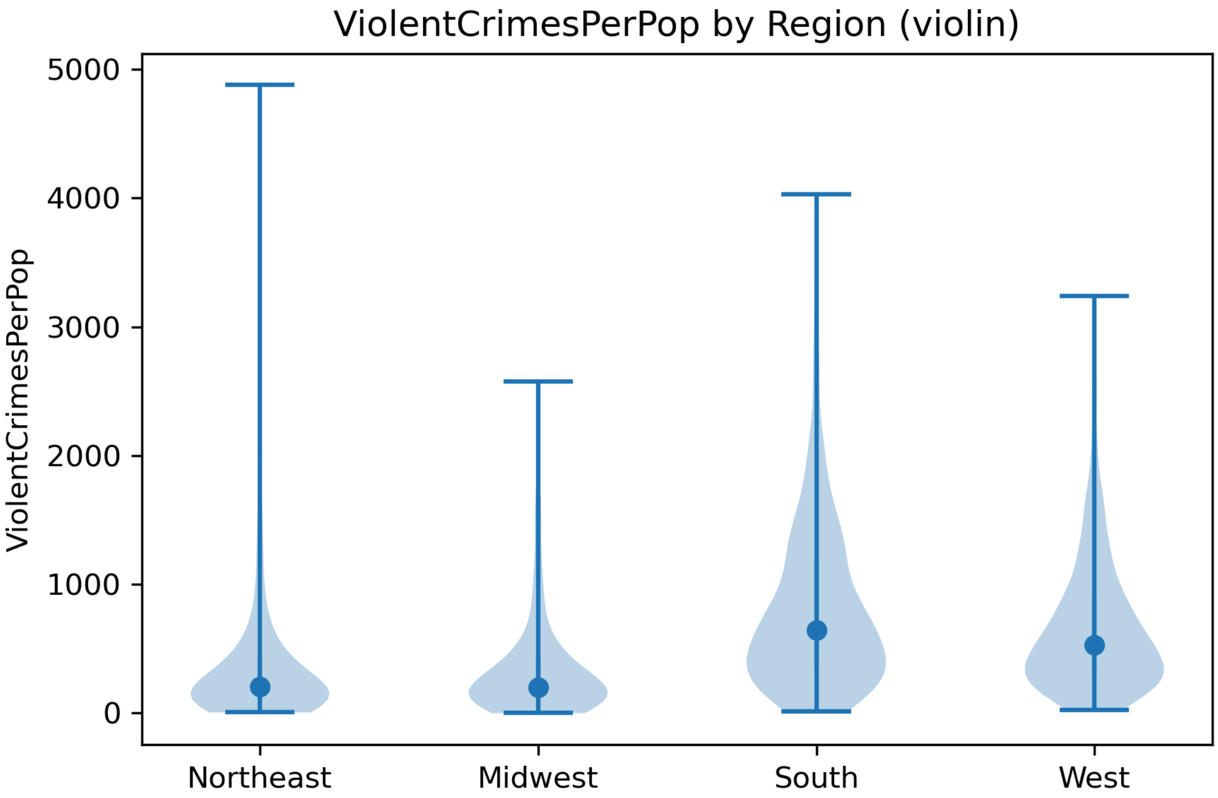

Figure 4. Distribution of ViolentCrimesPerPop by U.S. region.

3.Methods

3.1Data Splitting

The full dataset was randomly partitioned into three disjoint subsets: a training set (60%), a validation set (20%), and a test set (20%). The training set was used to fit model parameters, the validation set was used for model selection and hyperparameter tuning, and the test set was held out for final performance evaluation. This split ensures that model development and assessment are conducted on separate data, reducing the risk of overfitting and information leakage.

3.2 Data Preprocessing

Categorical features were encoded using a one-hot encoding scheme. Missing values in categorical variables were treated as a separate category and filled with the label ”missing”. Ordinal features were encoded using an ordinal encoder, with missing values imputed as −1, which is outside the valid range of observed ordinal levels and preserves the integer nature of these variables.All continuous features were standardized using a standard scaler to ensure zero mean and unit variance. Missing values in continuous variables were not imputed at this stage and were handled in subsequent modeling steps.

3.3 Handling Missing Values in Continuous Features

After preprocessing, missing value patterns were examined across the dataset. The data could be grouped into three distinct missingness patterns, corresponding to different subsets of continuous features with missing values. Notably, these patterns were consistent across the three sets, with the majority of observations belonging to a small number of well-defined patterns.

Given the limited number of missingness patterns, a **r**educed-feature modeling strategy was adopted to handle missing values in continuous features. For each missingness pattern, models were trained using only the subset of features observed for that pattern, allowing all available observations to be retained without imputing continuous variables. This approach avoids introducing potential bias from imputation while leveraging the structured nature of the missing data.

Missing pattern in training set	Pattern1 Rows: 191 Missing columns: None	Pattern2 Rows: 1004 Missing columns: 21	Pattern3 Rows: 1 Missing columns: 23
Missing pattern in validation set	Pattern1 Rows: 65 Missing columns: None	Pattern2 Rows: 334 Missing columns: 21	
Missing pattern in test set	Pattern1 Rows: 63 Missing columns: None	Pattern2 Rows: 336 Missing columns: 21	

Table 1. Missing value patterns across data splits.

3.4 Machine Learning Pipeline

Within this reduced-feature framework, five regression models were evaluated: Ridge Regression, k-Nearest Neighbors (kNN), Decision Tree, Random Forest, and XGBoost. These models were selected to represent a range of linear and non-linear approaches with varying capacity and inductive biases. Model hyperparameters were tuned using the validation set, and performance was assessed on the held-out test set.

In addition to the reduced-feature approach, XGBoost was also evaluated separately using the full feature set without explicit handling of missing values. XGBoost can natively handle missing values by learning optimal default directions for missing entries during tree construction, providing a useful comparison to the reduced-feature modeling strategy.

Model	Hyperparameters tuned (param_grid)

XGBoost(reduced feature model)	`learning_rate`: [0.03] ;
	`n_estimators`: [10000];
	`reg_alpha`: [0, 1e-2, 1e-1, 1, 10, 100];
	`reg_lambda`: [0, 1e-2, 1e-1, 1, 10, 100];
	`missing`: [np.nan];
	`max_depth`: [1, 3, 10, 30];
	`colsample_bytree`: [0.9];
	`subsample`: [0.66]
	Early_stopping_rounds: [50]

kNN Regressor(reduced feature model)	`knn__n_neighbors`: [3, 5, 11, 21];
	`knn__weights`: ["uniform", "distance"];
	`knn__p`: [1, 2]

Ridge Regression(reduced feature model)	`ridge__alpha`: [1e-3, 1e-2, 1e-1, 1, 10, 100];
	`ridge__fit_intercept`: [True, False];
	`ridge__random_state`: [0]

Decision Tree Regressor(reduced feature model)	`max_depth`: [None, 3, 5, 10, 20];
	`min_samples_split`: [2, 5, 10, 20];
	`min_samples_leaf`: [1, 2, 5, 10]

Random Forest Regressor(reduced feature model)	`n_estimators`: [200, 500];
	`max_depth`: [None, 10, 20];
	`min_samples_split`: [2, 5, 10];
	`min_samples_leaf`: [1, 2, 5];`max_features`: ["sqrt", 0.5, 1.0]

XGBoost(standalone)	`learning_rate`: [0.03] ;
	`n_estimators`: [10000];
	`reg_alpha`: [0, 1e-2, 1e-1, 1, 10, 100];
	`reg_lambda`: [0, 1e-2, 1e-1, 1, 10, 100];
	`missing`: [np.nan];
	`max_depth`: [1, 3, 10, 30];
	`colsample_bytree`: [0.9];
	`subsample`: [0.66]
	Early_stopping_rounds: [50]

Table 2. Models and hyperparameter grids used in the machine learning pipeline.

Model performance was assessed using RMSE and R², evaluated on the test set.

4 Results

All models substantially outperform the mean-prediction baseline (RMSE = 589.93). The standalone XGBoost model achieves the best overall performance, with the lowest RMSE and highest R² on the test set, while Random Forest and reduced-feature XGBoost show comparable but slightly weaker results. Overall predictive performance remains moderate, consistent with prior findings on this dataset.

Model	Hyperparameters tuned (param_grid)	RMSE	R2

XGBoost(reduced feature model)	learning_rate: 0.03 ;
	n_estimators: 10000;
	reg_alpha: [10; reg_lambda:0.1;
	missing: [np.nan];
	max_depth: 3; colsample_bytree: 0.9;
	subsample: 0.66
	Early_stopping_rounds: 50
		372.5162637433852	0.6012600941519646

kNN Regressor(reduced feature model)	knn__n_neighbors: 5; knn__weights: "distance"; knn__p: 2	401.5154308045312	0.5367625032485186

Ridge Regression(reduced feature model)	ridge__alpha: 10;
	ridge__fit_intercept: False;
	ridge__random_state: 0
		380.2262575006382	0.5845837940917527

Decision Tree Regressor(reduced feature model)	max_depth: 3;
	min_samples_split: 2;
	min_samples_leaf: 1
		414.6017194595276	0.5060745287068167

Random Forest Regressor(reduced feature model)	n_estimators: 200;
	max_depth: 20;
	min_samples_split: 10;
	min_samples_leaf: 1;
	max_features: "sqrt"
		369.5468385681643	0.6075916792606938

XGBoost(standalone)	learning_rate: 0.03 ;
	n_estimators: 10000;
	reg_alpha: [10; reg_lambda:0.1;
	missing: [np.nan];
	max_depth: 3; colsample_bytree: 0.9;
		369.1724479324817	0.6083863800428047

	`subsample`: 0.66 Early_stopping_rounds: 50		

Table 3. Comparison of regression model performance.

Both feature importance approaches identify a consistent set of highly influential predictors. Variables related to family structure—such as the proportion of children living in two-parent households and the proportion of children born to never-married parents—emerge as the most important features across both XGBoost gain-based importance and SHAP-based global importance. Demographic and housing-related variables, including racial composition, population density, and homelessness measures, also appear among the top predictors. While the exact rankings differ between the two methods, the substantial overlap suggests that the model's predictions are driven by a small number of robust and interpretable features rather than by unstable or idiosyncratic effects.

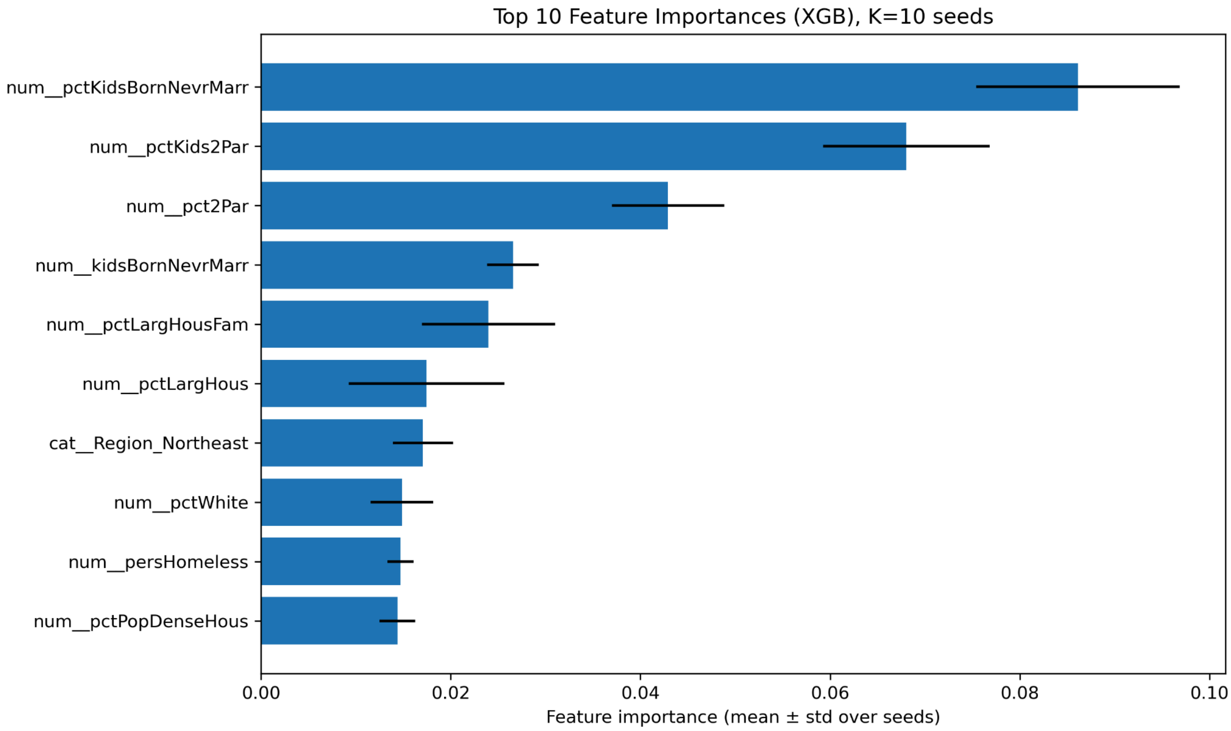

Figure 5. Top 10 feature importances from XGBoost (averaged over 10 random seeds). Bars show the mean feature importance across seeds, with error bars indicating the standard deviation, highlighting both the relative importance and stability of the most influential features.

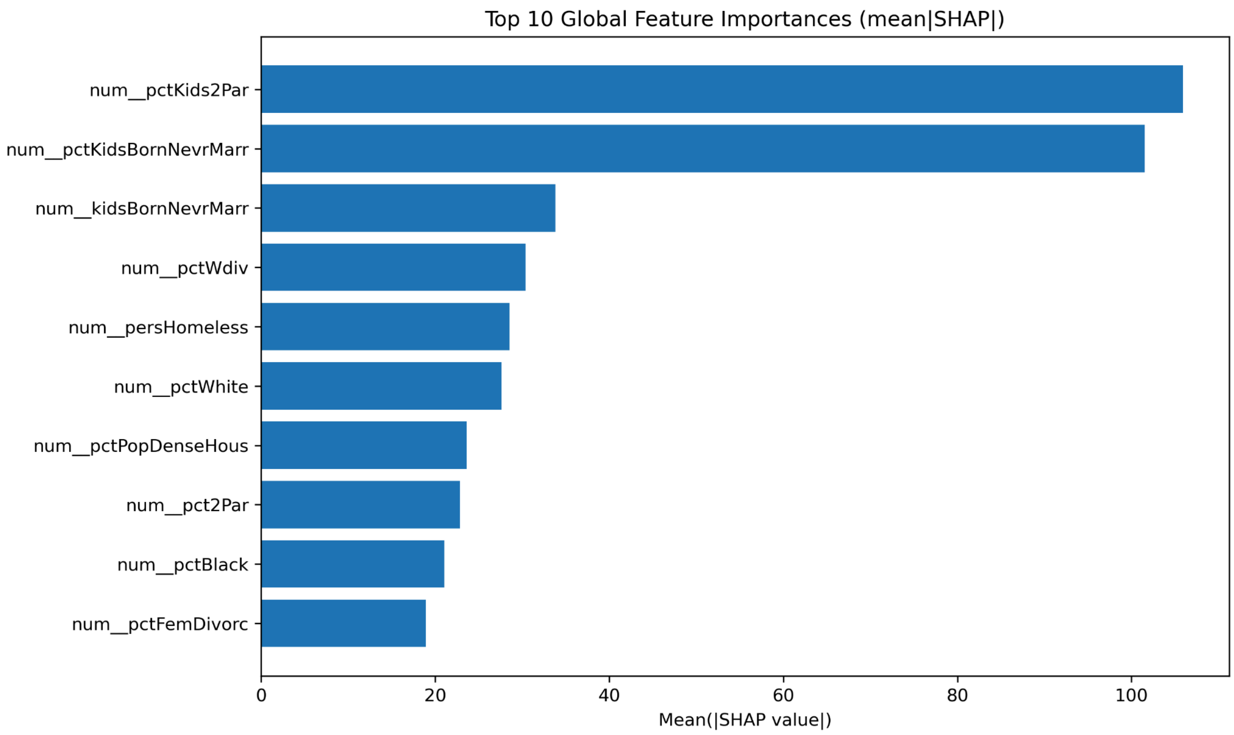

Figure 6. Top 10 global feature importances based on mean absolute SHAP values. Features are ranked by their mean absolute SHAP values, reflecting their average contribution to model predictions across all observations.

The two local SHAP explanations illustrate how the same set of influential features can affect predictions in different directions depending on community-specific values. For Sample #66, higher values of variables related to family structure and population density (e.g., *pctKidsBornNevrMarr*, *pctKids2Par*) contribute positively to the prediction, resulting in a crime rate above the baseline. In contrast, for Sample #27, strong negative contributions from similar family-structure indicators substantially reduce the predicted crime rate, highlighting the heterogeneity of feature effects at the individual-community level despite consistent global importance rankings.

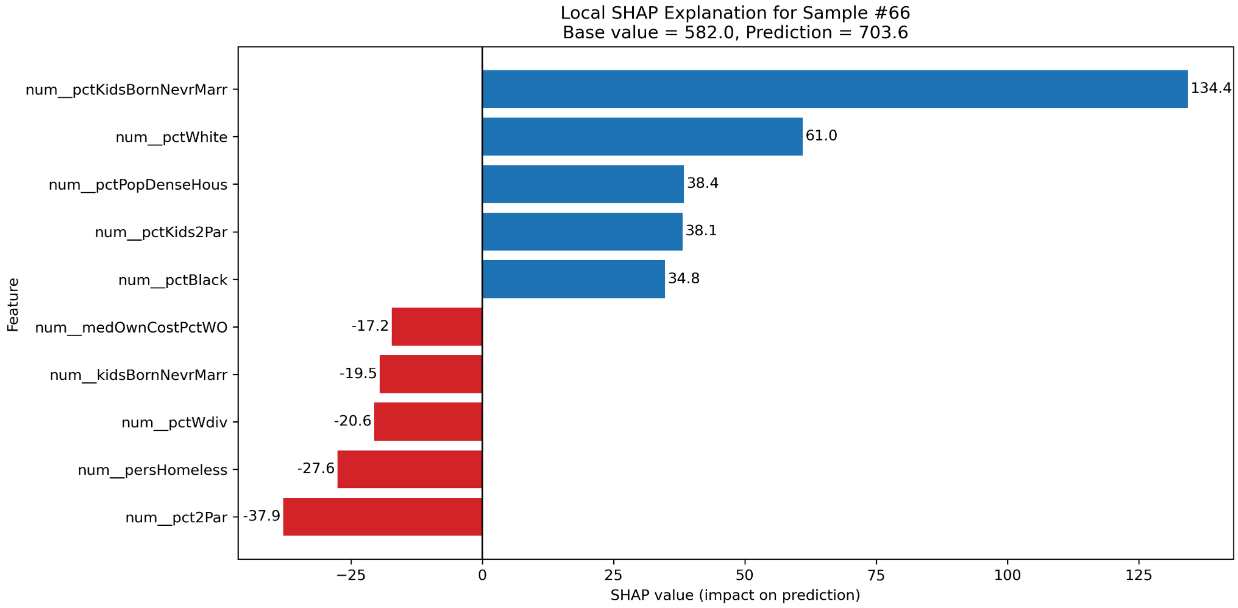

Figure 7. Local SHAP explanation for Sample #66 from the XGBoost model.

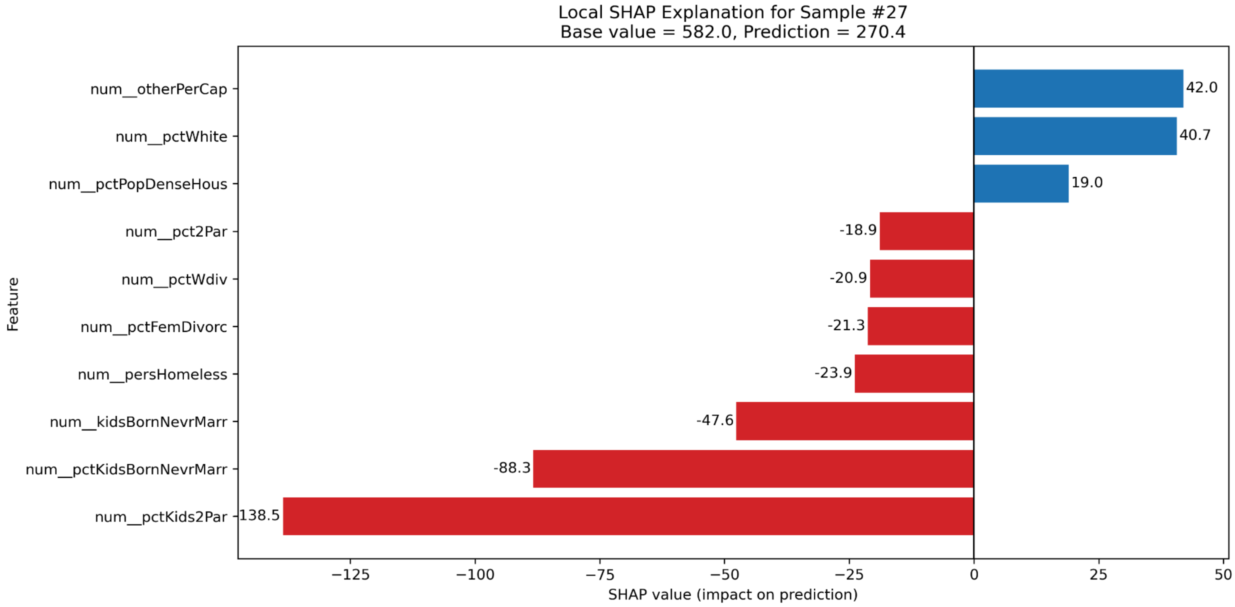

Figure 8. Local SHAP explanation for Sample #27 from the XGBoost model.

Feature importance and SHAP analyses further indicated that variables related to family structure, population composition, and housing characteristics were consistently influential, while local explanations revealed substantial heterogeneity in how these factors contribute to predictions across individual communities.

5. Outlook

Model performance could be improved by exploring more advanced imputation strategies for continuous features, rather than relying on reduced feature subsets. Additional models such as LightGBM or CatBoost may further improve predictive accuracy while handling missing values more effectively. From an interpretability perspective, interaction effects between key socio-demographic variables could be examined using SHAP interaction values or partial dependence plots. A key limitation of the current approach is the use of cross-sectional data, which cannot capture temporal changes in crime patterns. Collecting longitudinal crime, policy, or economic data would likely improve both predictive performance and interpretability.

6.References

[1]UCI Machine Learning Repository (1996) *Communities and Crime Unnormalized Data Set*. Available at: https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized
[2]Redmond, M.A. and Baveja, A. (2002) ‘A data-driven software tool for enabling cooperative information sharing among police departments’, *European Journal of Operational Research*, 141(3), pp. 660–678. https://doi.org/10.1016/S0377-2217(02)00131-9
[3]Chen, T. and Guestrin, C. (2016) ‘XGBoost: A scalable tree boosting system’, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

7.Github Repository

https://github.com/hanhuang-Rosie/data1030-project1/tree/main