Visualizing and Modeling United States Community Data against Crime Statistics

Annie Edmonds Harry Shield

1. Data Description

Our project utilizes a dataset that focuses on quantifying community diagnostics and crime statistics across communities in the United States. This dataset integrates information from three primary sources: socioeconomic data from the 1990 U.S. Census, law enforcement data from the 1990 Law Enforcement Management and Administrative Statistics (LEMAS) survey, and crime data from the 1995 FBI Uniform Crime Reporting (UCR) Program. The creators of this dataset included the most relevant attributes from the three sources, excluding any unrelated variables that had no connection to crime in U.S. communities. Overall, the combination of these three sources produces a dataset that is a length of 1994 instances, with a comparison of 127 features. Of the 127, five are non-predictive categorical values used for identifying U.S. states, counties, communities, and cross-validation numbers. The other 122 features are predictive measures that contain numerical data for making predictions.

The five non-predictive categorical values dataset provides geographic identifiers that allow for both broad and localized analysis of crime statistics. The 122 predictive measures include demographic variables such as age and race, as well as socioeconomic indicators, including unemployment rates and education levels. Additional personal data points include marital status, number of children, immigration status, English proficiency, and family dynamics. Law enforcement-specific variables include officers per capita, total LEMAS per population, and department demographics such as officer race. Additional data points cover law enforcement budgets, drugs seized, and overall crime rates per population.

In total, the dataset contains 127 variables, making data cleaning a challenging task. To manage this, selecting a subset of 25 features that includes a mix of predictive and non-predictive variables will enable us to adequately clean and create models, as well as visualizations, to represent the communities in the dataset. These chosen features are:

- state, county, community, communityname, population, householdsize, racepctblack, racepctWhite, racepctAsian, racepctHisp, agePct12t21, agePct12t29, agePct16t24, agePct65up, PctPersDenseHous, medIncome, PctPopUnderPov, PctForeignBorn, PolicOperBudg, ViolentCrimesPerPop, LandArea, PopDens, PolicPerPop, LemasTotalReq, NumInShelters

Each variable will have its own cleaning and processing pipeline, as outlined below. However, as most variables are continuous, they all may require transformation of different types. Widely, the dataset includes missing values that must be addressed during our data preparation stage to better equip the later stages of model production and feature visualization. The missing values are identified by a question mark within the dataset, which we will coerce to NaN values during the cleaning stage.

Most of the data are derived directly from the three mentioned sources; however, the creators of this dataset computed additional variables for better data understanding. For instance, the per capita violent crime variable is calculated using population data. There is also a sum of crime variable that aggregates major violent offenses in the U.S., including murder, rape, robbery, and assault.

The predictive values are represented on a normalized scale between 0 and 1, therefore will not need as much cleaning, other than to handle missing values. In order to normalize the values between 0 and 1 for consistency, the authors utilized an unsupervised, equal-binning method. The unsupervised aspect is defined as the data was only adjusted in accordance with that feature's values, without input from other columns or target values, such as violent crimes. The equal-binning aspect of the method sets the largest value at 1.00 and the smallest value at 0.00, then uses equal sized proportions to set the different values that fall between the minimum and the maximum. This alters how we can visualize and model our data, because we do not know what the initial numbers are from this data set, only the normalized ones. Therefore we can only make assumptions in comparison to the values found within each feature. It is also worth noting that this normalization does not help maintain relationships between values of attributes, meaning that comparing values for related variables, such as whitePerCap and blackPerCap for a community, would not yield noteworthy insights. This adds additional challenges when we are in the process of cleaning and preparing the dataset to make

meaningful comparisons between related variables and produce visualizations to demonstrate their relationships.

The non-predictive variables, state, county, community, and communityname use numeric codes and strings to represent their values. The state feature uses readily available Federal Information Processing Standards (FIPS) numeric codes. These have values ranging from 1 to 56 (skipping some numbers) to represent the 50 states. In the cleaning process, it may be helpful to create a dictionary to map the state names, as they are more useful in data visualization. This is similar to the county feature, which has a code specific to each state, using the same FIPS numeric codes. Although changing this to county names is less useful, because they are less recognizable from a visualization standpoint, this could be helpful, as they are more recognizable than numbers. Lastly, the community codes are the least beneficial, other than helping to provide a possible basis for missing data. If there is a communityname present, then a community code may be produced. As well as if the state FIPS code or county FIPS code is missing, the communityname might provide enough information to fix the missing values in these features.

2. Methods

Annie The goal of this project is to analyze the relationship between socioeconomic and demographic factors and the rate of violent crimes across different US communities. We aim to develop models that estimate the ViolentCrimesPerPop variable (the rate of violent crimes per population) based on specific community-level aspects. We strive to identify which features, socioeconomic and demographic factors contribute the most to violent crimes within different communities. We will use four different models for our project: linear regression, K-Nearest Neighbor regression, K-Means clustering, and a decision tree model.

For each of our models, we will need to prepare the data by cleaning data, handling missing values, transforming data types, and encoding categorical values. We will then complete an exploratory data analysis to understand the scale of our data, patterns, relationships, and any potential significant outliers. In this step, we will create visualizations and plots to demonstrate any major features and patterns easily. For the linear regression, K-Nearest Neighbor, and the decision tree models, the data will be split into training and testing sets (80% training and 20% testing). Since K-Means clustering does not predict a target variable, splitting the data is unnecessary. The next step is model training, which involves fitting the model to the training data. For linear regression, this entails minimizing the sum of squared errors between predicted and actual values. For K-Means, the algorithm iteratively assigns points to the nearest centroid and updates the centroids until convergence. For decision trees, the algorithm will recursively split features to reduce variance, producing terminal nodes that provide predictions for the future.

To produce the exploratory analysis and model each algo- Harry rithm, various libraries must be utilized within Python to ensure the reliability and validity of data modeling. Firstly, matplotlib is used to visualize histograms, line plots, and heat maps. It also ensures that data visualizations can be correctly labelled, titled, and customized to effectively encapsulate the correlation between different data variables. An extension for exploratory analysis is the usage of the seaborn package, which allows for further visualization. To make the algorithmic models, the scikit library is imperative. From the library sklearn.linear_model, LinearRegression is imported to create the linear regression model. From the library sklearn.neighbors, KNeighborsClassifier, and KNeighborsRegressor are imported for KNN regression. From the library sklearn.cluster, KMeans is imported to help with the kMC algorithm. Lastly, from the library sklearn.tree, DecisionTreeClassifier, and DecisionTreeRegressor are imported for decision tree modeling.

2.1. Linear Regression:

The linear regression model helps quantify how an independent variable (X) or variables affect a dependent variable (Y). It is utilized by assuming the interaction between variables is a linear relationship. By calculating a slope and intercept coefficient, as well as an error residual, once trained, the model can provide a predicted Y value from a new given X value. In our project, we use this algorithm to estimate the ViolentCrimesPerPop variable, which in this case is our dependent (Y) variable. By applying the linear regression model to specific demographic variables (X) as mentioned above, we can predict the quantity of violent crimes per population and determine how these variables impact the probability of violent crime in specific populations compared to others.

Analyzing a linear regression model, the R^2 value, which ranges from 0.00 to 1.00, provides insight into how well the regression line fits the data. It is calculated by dividing the variance explained by the model by the total variance within the data. A value closer to 1 indicates a higher likelihood of correlation between variables X and Y. Another critical metric is the root mean squared error (RMSE), which quantifies how far the model's predicted values fall from the actual observed values. Overall, it helps to create a more applicable analysis to real-world data by quantifying the accuracy of the model. Since the goal of linear regression is to minimize the SSE, using first-order conditions for optimization can also be a way to validate the model. Other options to expand the range of variables that the model can use to explain the variation in the data are maxmin normal-

ization, z-score normalization, logarithmic transformation, or inverse hyperbolic sine transformation.

2.2. K-Nearest Neighbor:

Annie The KNN regression algorithm predicts continuous numerical values by using the target values of similar data points. This algorithm computes the distance from the new point to each observation in the training dataset. It then identifies the k closest data points (k nearest neighbors) based on those distances and associates each of them with a known outcome value. Next, the algorithm computes the average of these k nearest outcome values, and this average becomes the predicted value for the new data point. This means that KNN regression predicts outcomes based on similarities among feature values, making it a beneficial tool for predicting quantities based on similar observations. This algorithm is valuable to our project as it enables us to predict the rate of violent crimes in a community by analyzing rates in comparable communities. These similarities are determined by factors such as population size, socioeconomic status, and demographic characteristics.

One important part of model validation for KNN regression is feature normalization and scaling. Since the distances in kNN depend on the size and scale of the variables, scaling one variable while leaving others unchanged often changes predictions. One solution is to use MaxMin normalization to reduce the effect of a change in values on the model's Harry performance. The KNN regression is validated using a training and test set, where some data is used for model training and the remaining data serves as a test to assess the model's closeness to the actual data points. These individually compute residuals, which come together as a sum of squared residuals (SSE), showing how well the model fits the dataset. These also help quantify the discrepancy between the i-th prediction and the actual value.

2.3. K-Means Clustering:

Annie Given the 1,994 instances in our dataset, the K Means clustering algorithm is particularly useful as it can handle very large datasets. The K-Means clustering algorithm starts with initialization and randomly selects k points as the centroids. Then, it calculates the distance of each observation to each centroid, assigns each data point to the closest centroid, and updates the centroid's value to the average of all assigned observations. These steps are repeated until convergence is reached. Although this algorithm does not target the ViolentCrimesPerPop variable, it can help to identify patterns and groupings in the data. By grouping data points with similar characteristics or values, we can determine the most similar communities. This will help us better understand crime patterns and how the demographic and socioeconomic profiles of communities affect crime rates. By clustering similar communities, we can broaden our insights to compare the average violent crime rate by cluster, instead of by each community individually.

Similar to KNN model validation, K-Means also benefits from feature normalization to ensure that all dimensions are similar in magnitude. MaxMin normalization can also be applied to enhance the model's validity. The K-Means Harry clustering algorithm can be validated through the calculation of the sum of squared error (SSE), which measures how well the model fits the data. kMC tries to minimize SSE bc this means the clusters are tighter. A scree plot is an important tool for determining the number of clusters (k) that are needed for the most efficient kMC fit. To form a scree plot, the SSE is plotted against the number of clusters (k), allowing us to visualize how adding or removing a cluster within the dataset affects the overall fit measure. To determine the optimal k value, the elbow of the scree plot offers valuable insight. When visualizing the SSE at k-1, k, and then k+1, there should be a significant drop. Once the drop is not as substantial, this signals there is no marginal benefit of adding an additional cluster. Therefore, increasing k further past this point would not improve the fit. In the event of no elbow on the scree plot, there are no discrete clusters present within the dataset. Another validation method involves examining the group statistics of the clusters to describe their condition based on their assignment. This can be done by using the .groupby('g_hat').describe() method, which organizes summary statistics for each feature of the cluster. This would provide further understanding of how the clusters differ from one another.

2.4. Decision Tree:

A decision tree model splits the data to minimize discrep- Annie ancies in outcomes within groups. This model consists of a set of decision nodes that represent decision points, a set of edges that represent data-driven choices, and a set of terminal nodes or outcomes at the bottom of the tree representing predictions. This method aims to build a predictive tree for data and future outcomes. This model predicts ViolentCrimesPerPop by analyzing community-level socioeconomic and demographic factors through the analysis of similar dataset groups. At each decision node, the algorithm chooses the feature and split that most reduces variance. This process continues until terminal nodes are reached, simultaneously creating branches that represent decisions that group homogeneous communities. For categorical variables, Gini impurity can be used to evaluate the homogeneity of the split node and the effectiveness of the split. However, for numeric variables, the aim of minimizing SSE can be an effective way to evaluate the splits.

A confusion matrix is a method of model validation for decision trees, as it calculates the model's accuracy and the proportion of correct predictions. Decision tree models

can also be defined by their tendency to be classified as overfitting or underfitting as an overall model. If a tree has too many branches, it ends up providing specific data to which few of the data sets refer. This indicates that the model has overfit the dataset, making it less applicable to new data points due to its specificity. On the other hand, if a model has insufficient branches and the data is too broad, it can be said to have underfit the dataset. As a result, a new data point is unlikely to be correctly fitted because of the model's broad nature. It is crucial to strike a balance between overfitting and underfitting to ensure that models accurately fit the data and, consequently, can effectively predict future data. There are multiple methods that can be used for decision trees to avoid overfitting and underfitting. Truncating the tree to limit its depth, setting a lower bound on the impurity for a terminal node, and limiting the number of cases at a terminal node can help prevent overfitting. Whereas, programming the tree to avoid splits that make the outcome populations too pure would help prevent underfitting.

Harry