CSE:587 DIC Project Report (Phase 2)

1. Akanksh Gatla (akankshg) 50465101
2. Gnana Abhinay Vadlamudi (gnanaabh) 50496402
3. Sai Teja Chittumalla (schittum) 50496091

Why is it a significant problem?

Background:

Crime is a pervasive social issue with far-reaching implications for the safety and well-being of communities. Analyzing crime data allows us to gain insights into the evolution of criminal activity, understand its regional disparities, and potentially inform policy decisions. The Unified Crime Reporting Statistics, compiled by the U.S. Department of Justice and the Federal Bureau of Investigation, provides a comprehensive dataset covering a wide range of years (1960 to 2019) and various crime categories.

Property crime, including burglary, larceny, and motor-related crimes, poses a significant financial and emotional burden on victims and communities. Violent crimes, such as assault, murder, rape, and robbery, endanger lives and impact the overall security of areas. Therefore, comprehensively studying these crime categories can provide valuable insights into their nature, causes, and possible preventive measures.

2. Data Sources :

Dataset used in this project - 'https://corgis-edu.github.io/corgis/csv/state_crime/'

From the Unified Crime Reporting Statistics and under the collaboration of the U.S. Department of Justice and the Federal Bureau of Investigation information crime statistics are available for public review. The following data set has information on the crime rates and totals for states across the United States for a wide range of years. The crime reports are divided into two main categories: property and violent crime. Property crime refers to burglary, larceny, and motor related crime while violent crime refers to assault, murder, rape, and robbery. These reports go from 1960 to 2019.

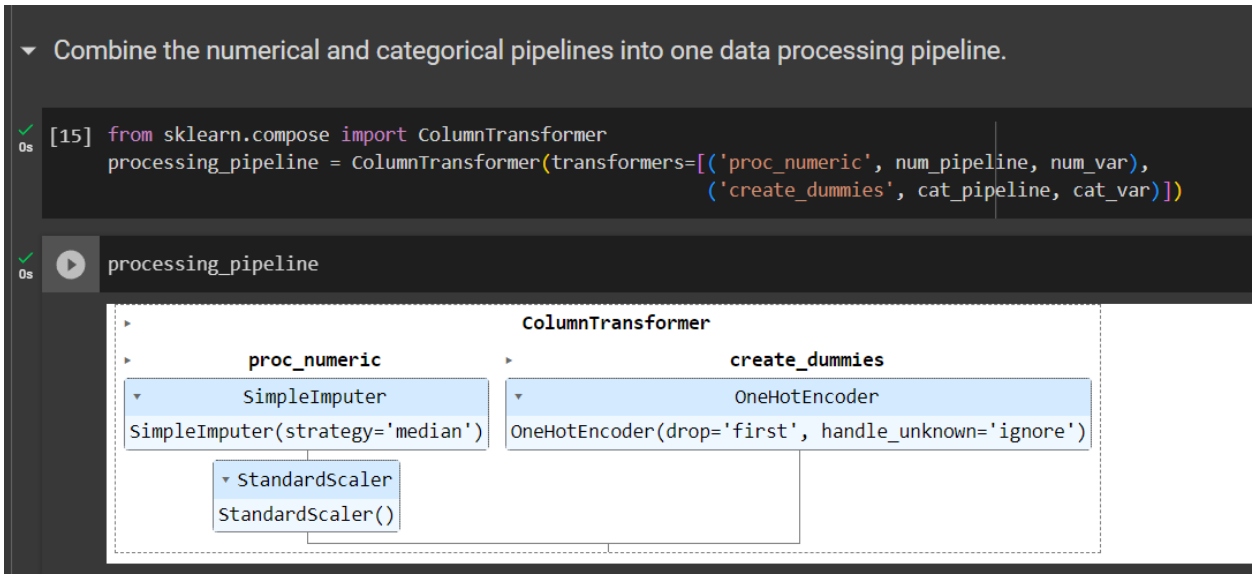
1. Data Description:
   The dataset contains information on crime rates and totals in different states and years. Columns include State, Year, Population, Rates (property and violent), and Totals (property and violent for various crime types).

2. Data Preparation:
   Selected a subset of columns as features (X) for the machine learning model.
   Defined the target variable (y) as the sum of specific columns representing different types of crime totals.

3. Train-Test Split:
   Used train_test_split from scikit-learn to split the data into training and testing sets.

Records in training data: 2,180
Records in test data: 935

4. Created Pipeline using one hot encoder to handle different types of string or missing errors on column transformers.



A. Linear Regression:

a. Why Linear Regression?

1. **Interpretability:**

Linear Regression provides a clear interpretation of the relationship between the independent variables (features) and the dependent variable (target).

2. **Simplicity:**

It's a good starting point, especially when exploring the dataset or as a baseline model before considering more complex algorithms.

3. **Assumptions:**

Assumptions like normality of residuals and homoscedasticity can be assessed and addressed if needed.

b. How applied?

Model Training:

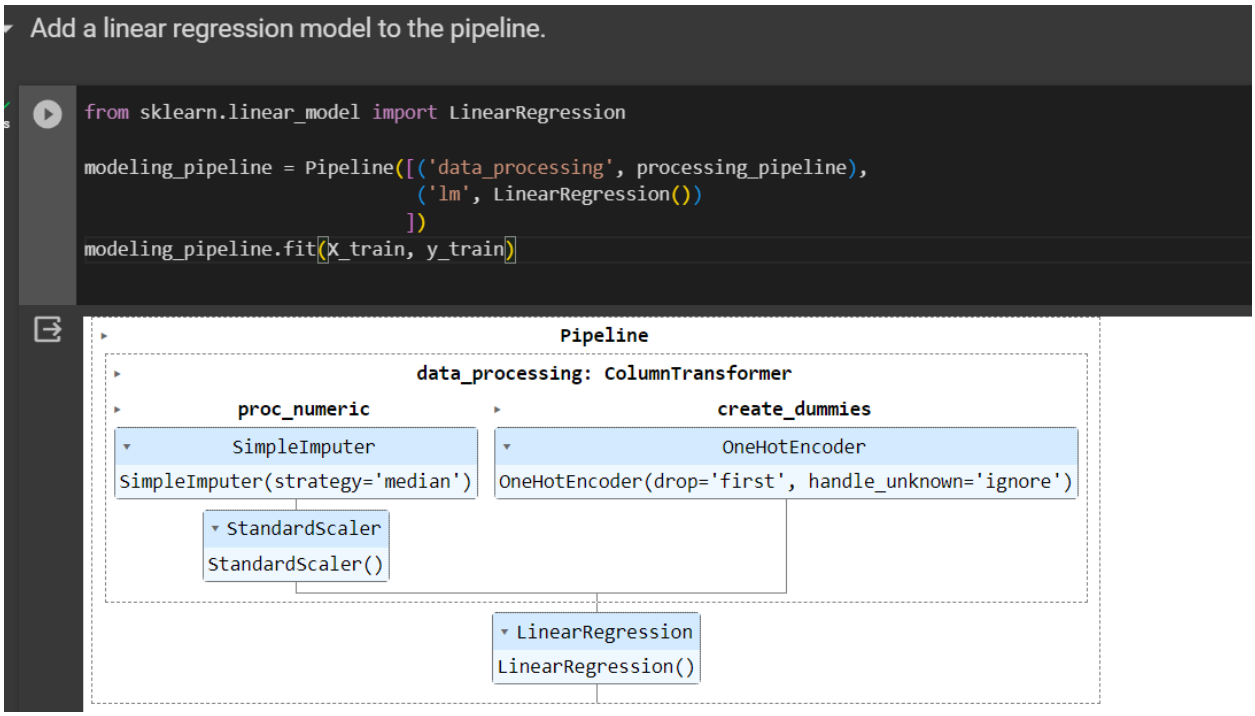
Utilized the scikit-learn library to create a Linear Regression model.

Fitted the model to the training data, allowing it to learn the relationships between features and the target variable.

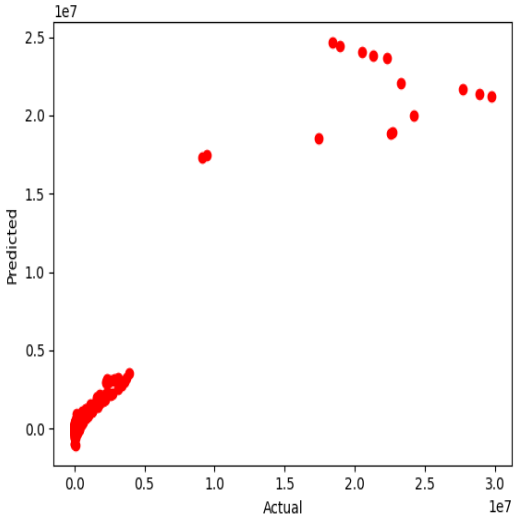
Expectations:

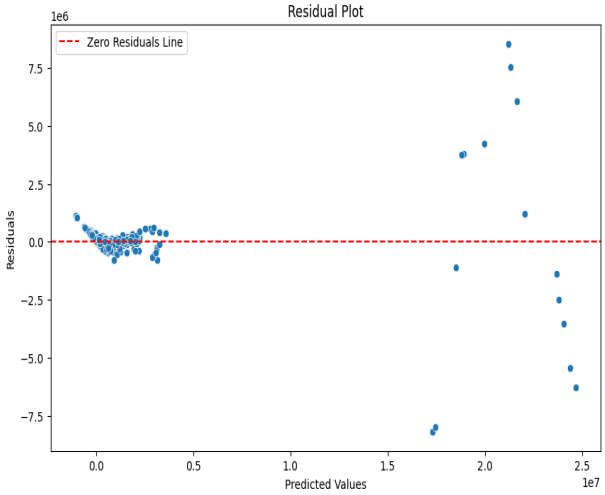
Coefficients from the Linear Regression model offer insights into the importance and direction of each feature's impact on crime totals.

Linear Regression is chosen for its interpretability, simplicity, and suitability for predicting continuous target variables. It serves as a foundational model in the initial stages of exploring the relationship between crime features and totals. The choice is made with an awareness of the model's assumptions and a willingness to refine the approach based on insights gained during the modeling process.



As we have already created the pipeline using that we just add it to our model. Next we predict the model and note the R^2 and mse values .
R^2 value for Linear Regression is : 93.37549997489647
MSE : 503108771106.021
RMSE: 709301.6079962184





B. **Ridge Regression**

Ridge Regression is chosen to address potential overfitting and multicollinearity in the crime prediction model. The iterative exploration of alpha values provides a nuanced understanding of how regularization impacts feature coefficients and overall model performance. This approach aims to enhance the interpretability and generalization of the predictive model.

a. Why Ridge Regression?

Regularization:

1. Ridge Regression is selected for its regularization property, which helps prevent overfitting by penalizing large coefficients. In the context of crime prediction, it's common to have multiple features that might be correlated. Ridge Regression can handle multicollinearity by shrinking the coefficients.
2. Alpha Parameter: The choice of Ridge Regression involves tuning the regularization parameter (alpha).The code explores various alpha values to understand the impact of regularization strength on model performance and coefficients.

Comparison with Linear Regression:

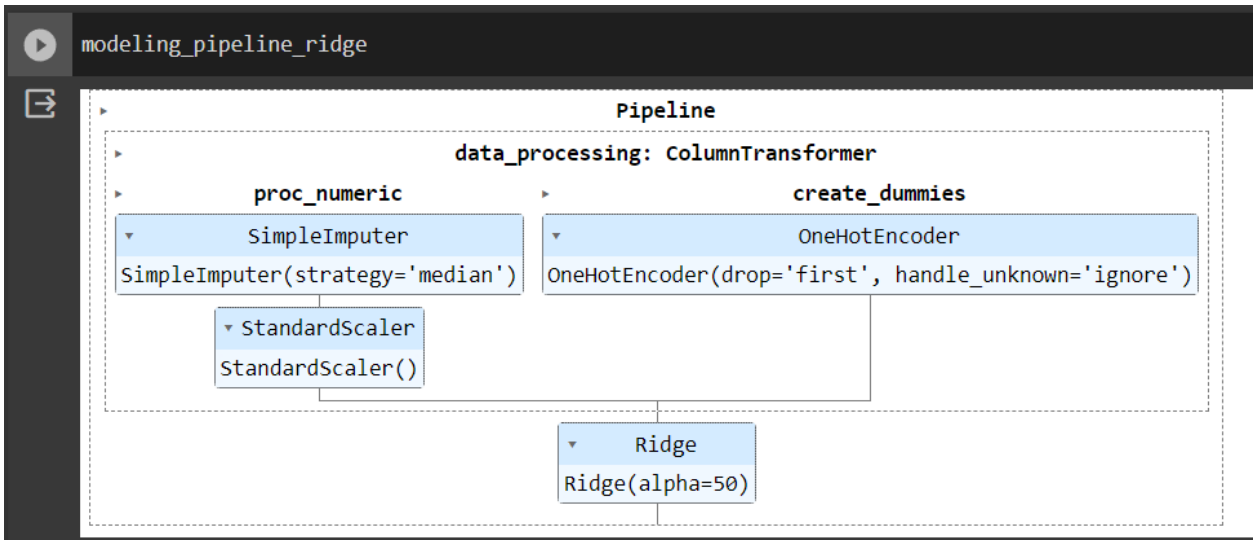
Ridge Regression is used as an extension or improvement upon simple Linear Regression. It addresses some of the limitations of Linear Regression, making it a more robust model in scenarios with potential multicollinearity.

b. How Applied:

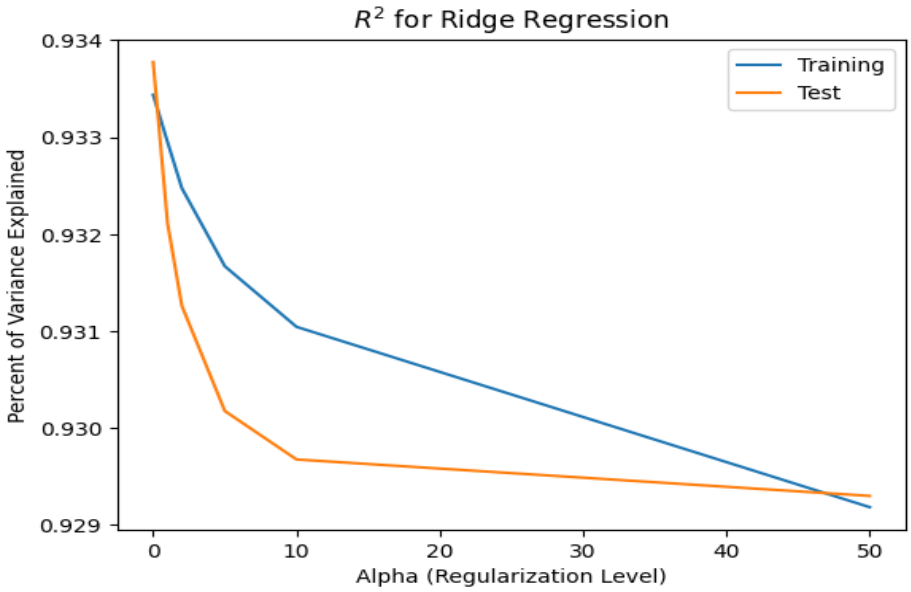
Pipeline Setup:

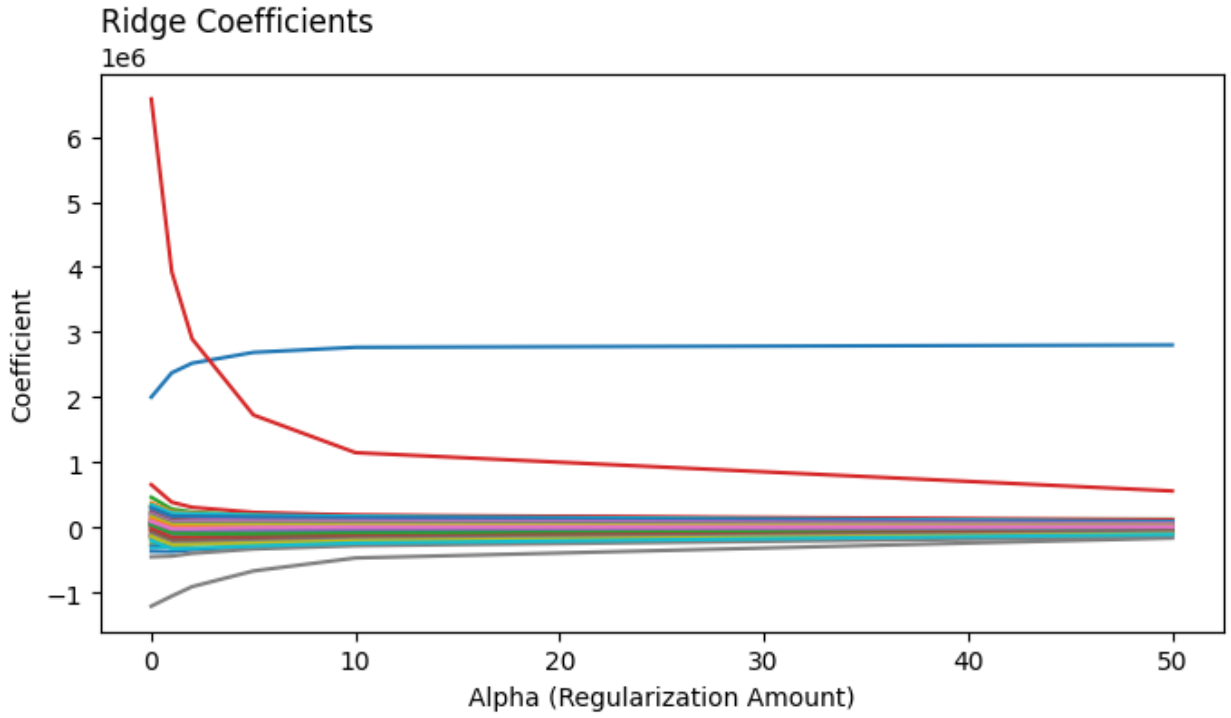
The Ridge Regression model is integrated into a scikit-learn pipeline along with a data processing step.

Model Training:
The pipeline is trained on the training set (X_train and y_train). The regularization parameter (alpha) is set to 100 initially, and the model is trained.
Grid Search Over Alphas:
A loop is conducted over a range of alpha values (0, 1, 2, 5, 10, 50) to explore different levels of regularization. For each alpha, a Ridge Regression model is trained, and coefficients, training score, and test score are recorded.
Results Visualization:
Coefficients for each alpha are visualized to observe how regularization affects feature importance. The training and test scores are plotted against different alpha values to understand the model's performance under different levels of regularization.
Expectations:
Overfitting Control:
Ridge Regression is chosen with the expectation that it will help control overfitting, especially in the presence of correlated features.
Alpha Tuning:
The iteration over different alpha values provides insights into the optimal level of regularization for the specific problem.
Model Evaluation:
The visualization of training and test scores over different alphas aids in assessing model performance and choosing a suitable regularization parameter.



R^2 for this model : 92.92975392202159





C. **Random Forest**

Random Forest is chosen for its ability to handle complex relationships, provide insights into feature importance, and offer robustness to overfitting. The choice is made with an awareness of the specific characteristics of the dataset and the problem at hand, aiming to enhance predictive performance and gain meaningful insights into the factors influencing crime totals.

a. Why Random Forest?

Complex Relationships:

Random Forest is chosen when the relationships between features and the target variable are expected to be more complex than what a linear model can capture.

It can handle non-linear relationships and interactions between features effectively.

Robustness to Overfitting:

Random Forest is less prone to overfitting compared to individual decision trees. The ensemble nature of Random Forest helps generalize well to unseen data.

Ensemble Method:

By combining multiple decision trees, Random Forest reduces the risk of individual trees making biased predictions, leading to a more robust model.

b. How Applied:

Exploration of Interactions:

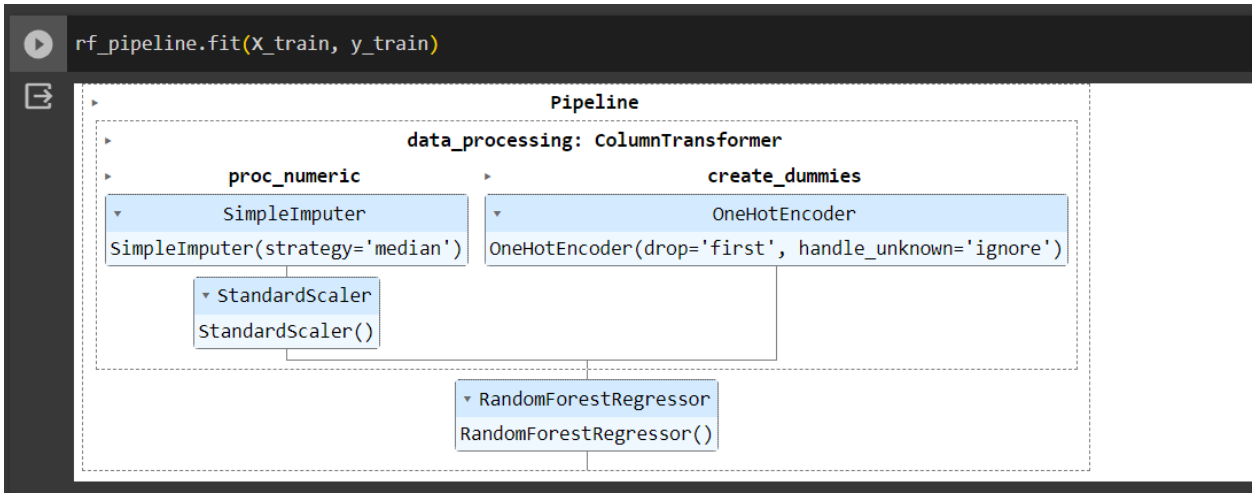
Random Forest is expected to capture interactions between different types of crimes (property vs. violent) and their rates more effectively than a linear model.

Hyperparameter Tuning:

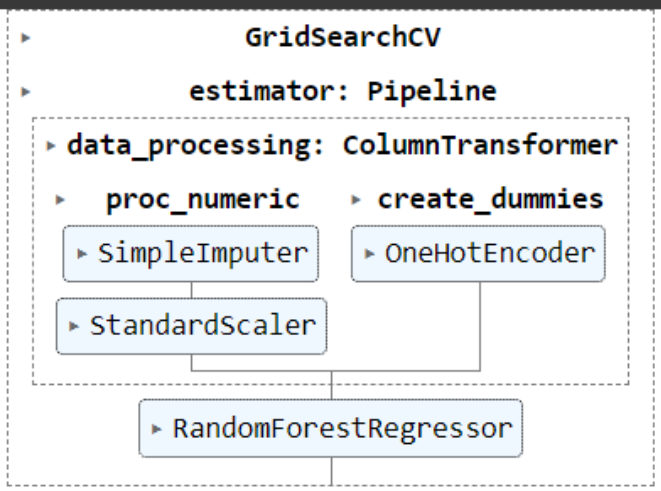
Utilized GridSearchCV to find the optimal hyperparameters for the Random Forest model.

c. Expectations:

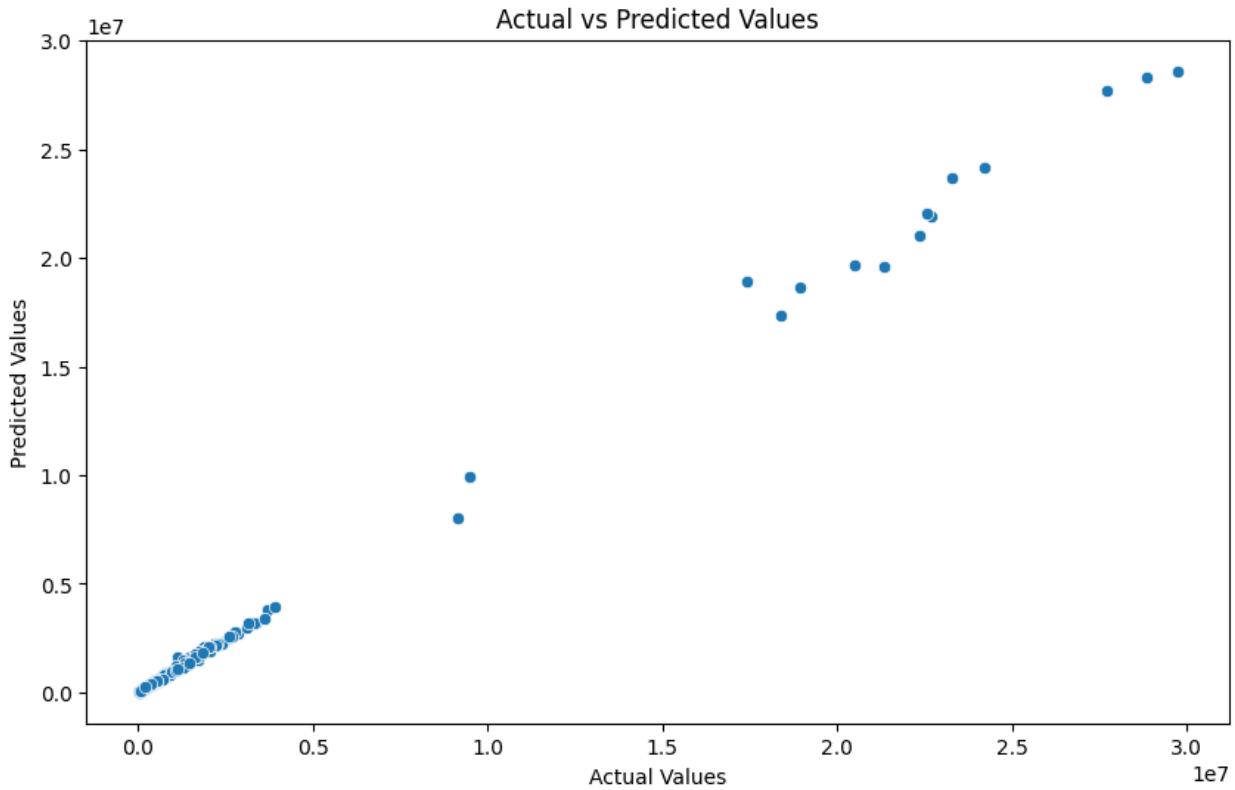
Extracted and analyzed feature importance to gain insights into which features contribute most to the prediction of crime totals. Learning: Understanding the relative importance of different crime-related features in the context of the problem.

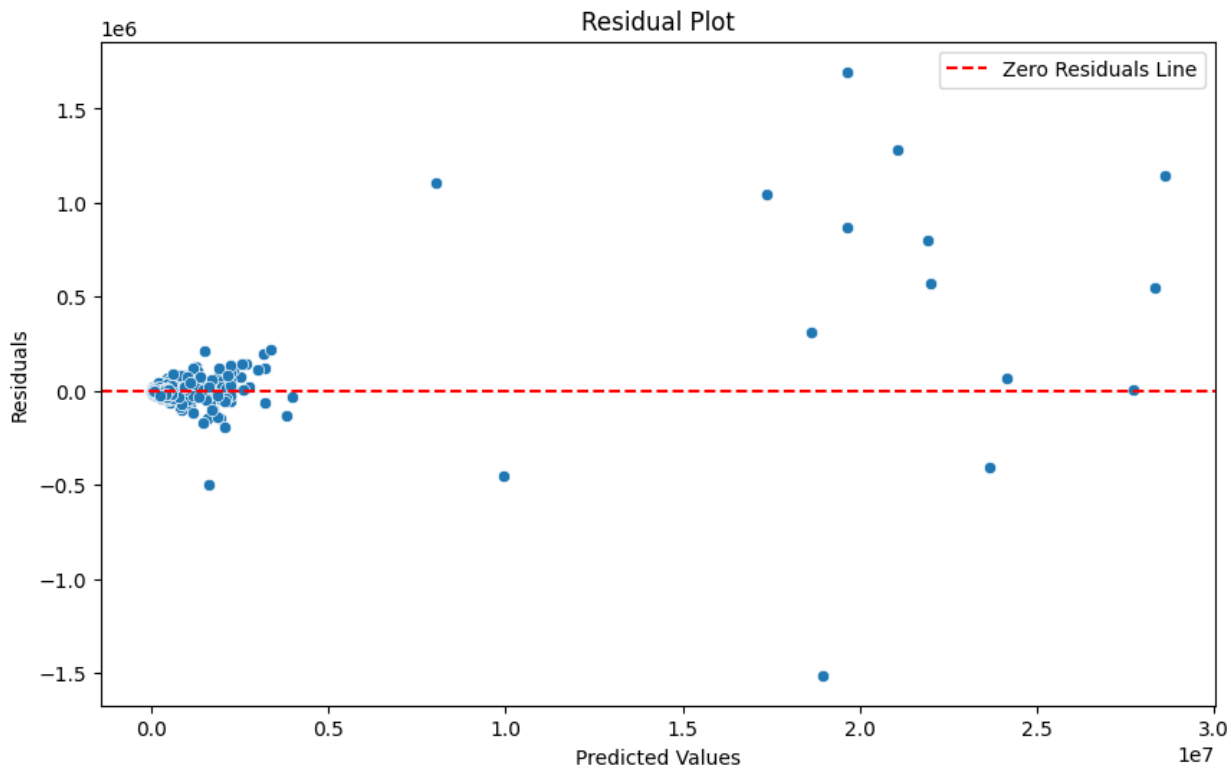


```
[35] grid_search = GridSearchCV(rf_pipeline, param_
     grid_search.fit(X_train, y_train)
```



Best Hyperparameters: {'rf__max_depth': 10, 'rf__min_samples_leaf': 1, 'rf__min_samples_split': 2, 'rf__n_estimators': 100}
Mean Squared Error on Test Set: 14977605963.532991
R-squared on Test Set: 99.80278787256422





D. **Decision Tree**

Decision Tree Regression is chosen for its ability to capture non-linear patterns, provide feature importance insights, and offer an intuitive decision-making process. The decision to use this model is made with an awareness of its strengths and considerations, and the evaluation metrics help assess its performance in the context of predicting crime totals.

a. Why Decision Tree?

Non-linearity in Relationships:

Decision Tree models are capable of capturing non-linear relationships between features and the target variable. Unlike Linear Regression, they can model complex interactions.

Handling Non-Normal Distributions:

Decision Trees don't assume a particular distribution of the data. This makes them robust to non-normal distributions, which might be present in crime rate data.

Intuitive Decision Making:

The decision-making process of a Decision Tree is easy to understand and interpret, making it valuable for providing insights into the factors influencing crime totals.

b. How applied?

Pipeline Integration:

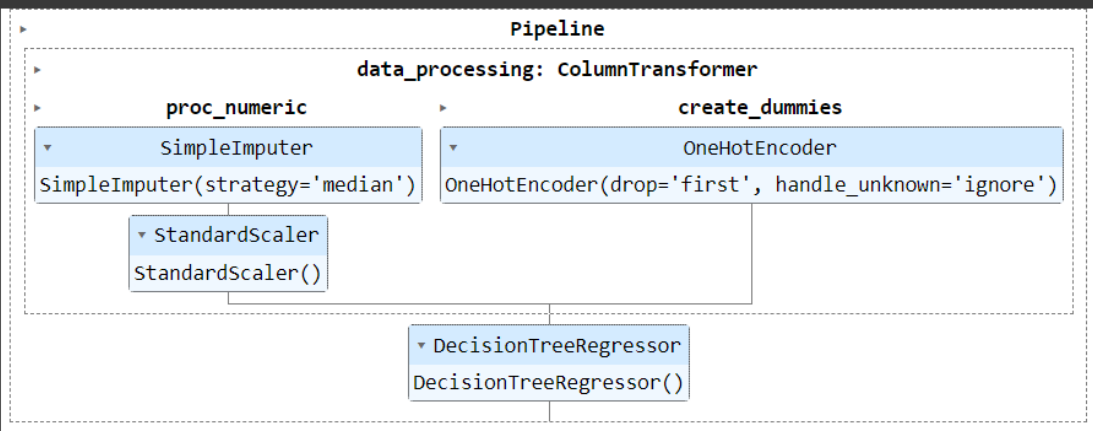
Integrated the Decision Tree model into a processing pipeline, allowing for consistent and reproducible data transformations.

Model Training:
Utilized the DecisionTreeRegressor from scikit-learn as the regression algorithm within the pipeline. Trained the model on the training set to learn the relationships between the features and the target variable.

Evaluation Metrics:
Used common regression metrics like R-squared, Mean Absolute Error (MAE), Mean Squared Error (MSE), Median Absolute Error, and Root Mean Squared Error (RMSE) to evaluate model performance.

c. **Expectation**
Anticipate gaining insights into the importance of different features in predicting crime totals. Expect Decision Tree Regression to capture non-linear patterns in the data, which may be present in crime rate variations

```
modeling_pipeline = Pipeline([('data_processing', processing_pipeline),
                              ('reg', DecisionTreeRegressor())
                             ])
modeling_pipeline.fit(X_train, y_train)
```




```
Mean_absolute_error:  43597.15294117647
Mean_squared_error:  66589717483.71551
Median_absolute_error:  8388.0
Root_Mean_Squared_error:  258049.83527163026
R_Square :  99.12320434371924
```

E. **KNeighborsRegressor**

KNeighborsRegressor was chosen for its non-linear nature and adaptability to complex data patterns. The application of this model was aimed at capturing localized relationships within the crime dataset, and the evaluation process provided insights into its performance and trade-offs.

a. Why KNeighborsRegressor?

Non-Linear Relationships:

KNeighborsRegressor is a non-linear model, which makes it suitable for capturing complex relationships in the data. Unlike Linear Regression, it does not assume a linear relationship between the features and the target variable.

Localized Predictions:

KNeighborsRegressor makes predictions based on the values of the k-nearest neighbors in the feature space. This can be advantageous when there are localized patterns in the data that may not be captured well by a global model like Linear Regression.

Adaptability to Data Distribution:

KNeighborsRegressor adapts well to the underlying data distribution without making strong assumptions about the functional form of the relationship. It can handle cases where the relationship between features and the target variable is not strictly linear.

b. How Applied:

Pipeline Integration:

Incorporated KNeighborsRegressor into a modeling pipeline along with data processing steps.Utilized a pipeline for consistency in preprocessing and modeling, ensuring a streamlined and reproducible workflow.

Model Training:

Fitted the KNeighborsRegressor model to the training data, allowing it to learn the relationships between features and the target variable.

c. Expectations:

Capturing Localized Patterns:

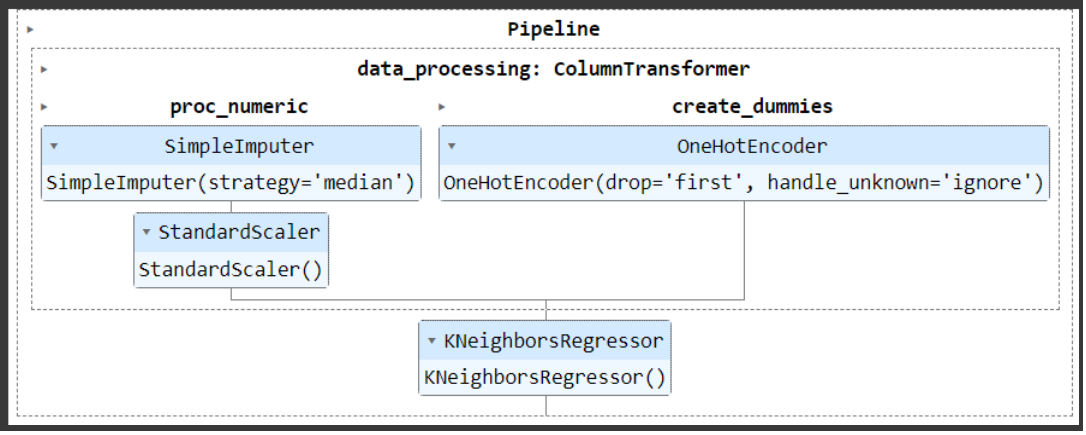
Expected KNeighborsRegressor to capture localized patterns in the data, which might not be effectively captured by a linear model.

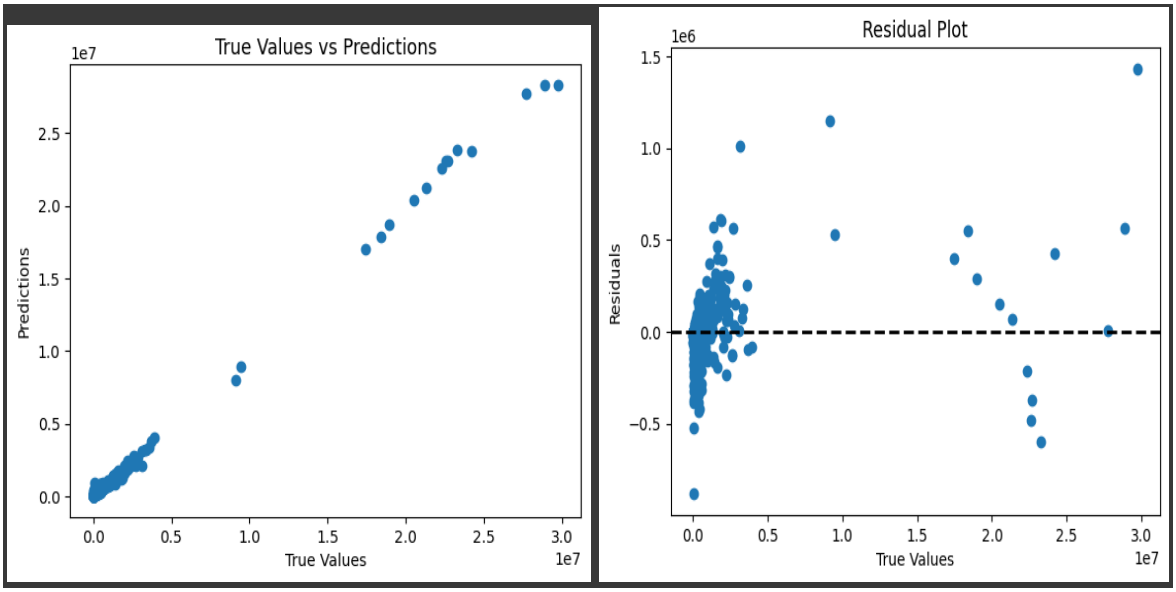
Adaptation to Non-Linearity:

Expected the model to perform well in cases where the relationships between features and crime totals are non-linear.

```
from sklearn.neighbors import KNeighborsRegressor

modeling_pipeline = Pipeline([('data_processing', processing_pipeline),
                              ('reg', KNeighborsRegressor())
                             ])
modeling_pipeline.fit(X_train, y_train)
```





R_Square : 99.78463031575497

F. **SVR**

Support Vector Regression (SVR) was chosen for its ability to handle non-linear relationships and provide flexibility through parameter tuning. The application involved constructing a pipeline, tuning hyperparameters, training the model, and evaluating its performance with various metrics. The expectations were centered around capturing complex patterns in the crime data, and the learnings involved insights gained from the model's behavior and performance metrics.

Why SVR?

Non-Linearity:

SVR is a powerful choice when dealing with non-linear relationships between features and the target variable. Linear Regression assumes a linear relationship, while SVR can capture more complex patterns in the data.

Flexibility:

SVR can handle high-dimensional data and is effective in cases where the relationship between features and the target is not straightforward. It is particularly useful when the data exhibits non-linear and intricate patterns that linear models might struggle to capture.

Kernel Trick:

The use of kernels in SVR allows it to transform the input features into higher-dimensional spaces, potentially making it more effective in capturing non-linear relationships.

Tuning Parameters:

SVR provides tuning parameters like C and gamma, allowing for flexibility in controlling the trade-off between model simplicity and fitting to the training data.

How Applied?

Parameter Tuning:

Tuned SVR hyperparameters (C and gamma) to find the optimal configuration for the specific dataset. This involves adjusting the regularization parameter (C) and the kernel coefficient (gamma) to achieve the best balance between bias and variance.

Model Training:

Fitted the SVR model to the training data, allowing it to learn the complex relationships between the selected features and the target variable.

Expectations:

Non-Linear Patterns:

Anticipated that SVR would outperform linear models by capturing non-linear patterns in the crime data.

Tuning Impact:

Tuning hyperparameters would impact model performance.

Learning: Analyzed how different values of C and gamma affected the model's ability to generalize.

Insights into Residuals:

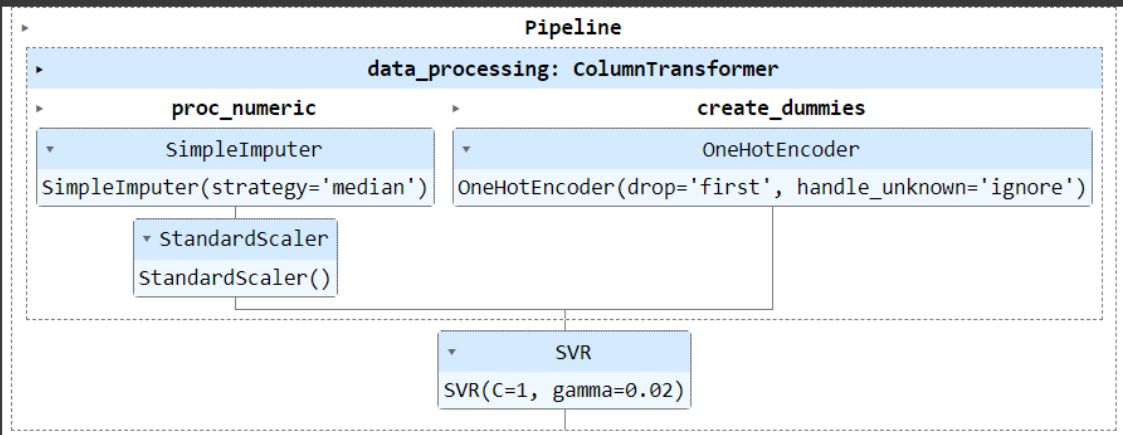
Studied the residuals to understand where the model performed well and where it struggled.

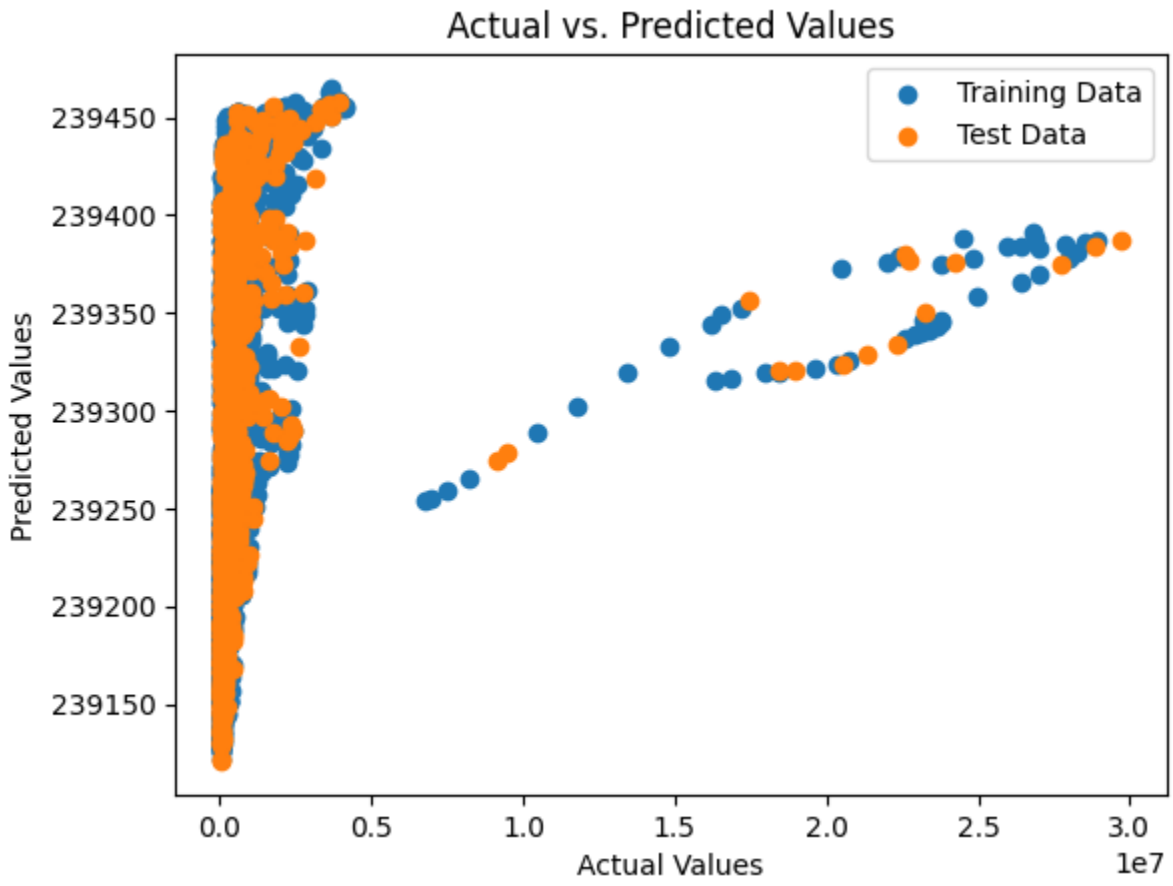
Learning: Extracted insights from residuals to potentially guide further model improvement or feature engineering.

SVR

```
from sklearn.svm import SVR

modeling_pipeline = Pipeline([('data_processing', processing_pipeline),
                              ('reg', SVR(C=1,gamma=0.02))
                             ])
modeling_pipeline.fit(X_train, y_train)
```





```
Mean_absolute_error:  636912.5129617485
Mean_squared_error:  7845036420979.6875
Median_absolute_error:  172843.93193509706
Root_Mean_Squared_error:  2800899.216498103
R_Square :  -3.2966367361669136
```

G. **Lasso Regression**

Lasso Regression is chosen for its ability to handle high-dimensional datasets, promote sparsity, and prevent overfitting. The expectations include improved feature selection, regularization benefits, and insights into the impact of different crime rates on total crime predictions. The learning process involves interpreting the model's coefficients and assessing its overall performance on both training and test sets.

a. Why Lasso Regression?

Feature Selection:

Lasso Regression is particularly useful when dealing with datasets with a large number of features. The L1 regularization term in Lasso introduces sparsity in the coefficients, effectively performing feature selection by pushing some coefficients to exactly zero. This can help in identifying and focusing on the most influential features, potentially improving model interpretability and generalization.

Handling Multicollinearity:

Lasso Regression is effective in handling multicollinearity, a situation where features are correlated with each other. By introducing sparsity, Lasso tends to pick one of the correlated features and shrink the coefficients of the others towards zero.

Preventing Overfitting:

Lasso's regularization term helps in preventing overfitting by penalizing large coefficients.

It provides a balance between fitting the training data well and keeping the model simple enough to generalize to new, unseen data.

b. How Applied:

Pipeline Integration:

Lasso Regression is incorporated into a modeling pipeline along with a data processing pipeline (processing_pipeline). The data processing pipeline likely includes steps such as scaling, encoding, and handling missing values to prepare the data for modeling.

Model Initialization:

The Lasso model is initialized with specific hyperparameters, including the regularization strength (alpha) and maximum number of iterations.

Training and Evaluation:

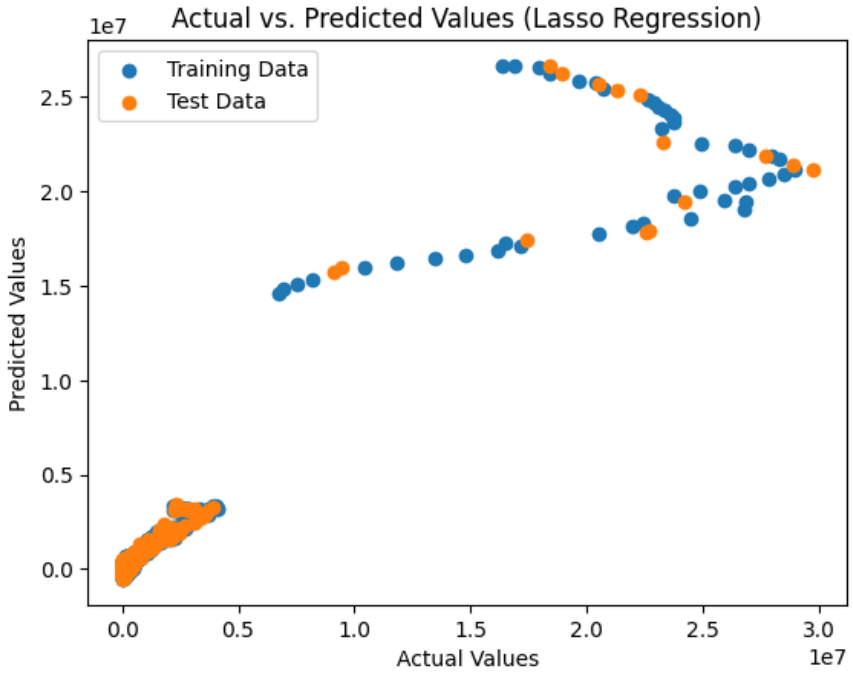
The modeling pipeline is fitted to the training data. Model performance is evaluated on both the training and test sets using the R-squared score.

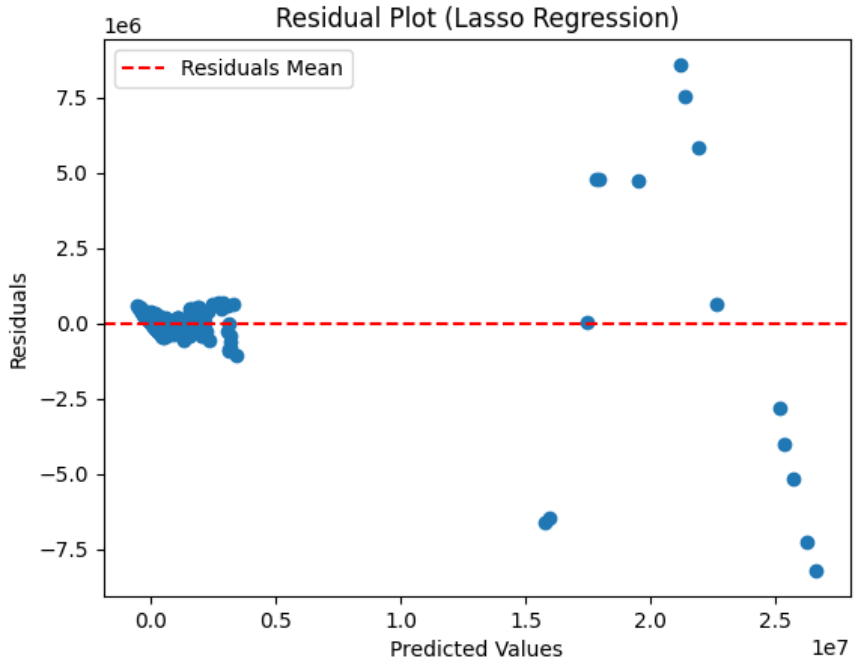
c. Expectations:

Sparsity and Feature Selection:

Expectation is that Lasso will promote sparsity in the coefficient estimates, leading to the selection of a subset of important features. This can reveal which crime rates or totals have a more significant impact on the prediction of total crime.

Regularization and Generalization:
Expectation is that Lasso's regularization will prevent overfitting, allowing the model to generalize well to new, unseen data. The balance between fitting the training data and simplicity introduced by the regularization term is a key expectation.





References:

1. https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/#:~:text=Lasso%20regression%20is%20a%20regularization,i.e.%20models%20with%20fewer%20parameters).
2. https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/
3. https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2