Analysis on Potential Contributing Factors of Violent Crimes in US Counties

DATA 583: Advanced Predictive Modeling

March 23, 2024

Jinxin Wang	95691002
Nan Tang	48735633
Xing Xu	42946970

Contents

Abstract

This report presents a comprehensive investigation into the multifaceted factors contributing to violent crimes across U.S. counties, employing a multi-feature dataset sourced from Kaggle which encompasses socioeconomic, demographic, public safety, and technological variables of 2,525 counties. The analysis, grounded in advanced predictive modeling techniques such as LightGBM, Linear Regression with LASSO regularization, Decision Trees, Random Forests, and Support Vector Machine Regression, aims to uncover the complex relationships between these factors and violent crime rates. Our findings corroborate the hypothesis that the LightGBM model outperforms others in predictive accuracy, thereby providing an understanding of crime dynamics. The study identifies key predictors of violent crime rates, including the number of law enforcement officers, the demographic composition, and socioeconomic indicators like per capita income and educational attainment levels. The report concludes with recommendations for policymakers and residents, suggesting a more targeted approach toward allocating resources and designing interventions to mitigate violent crimes effectively. Future avenues of research are proposed to enhance the predictive accuracy and applicability of violent crime rate models.

Keywords: Violent Crime Rate, Predictive Modeling, Machine Learning, LightGBM, U.S. Counties

1 Introduction

According to the FBI's data [1], the US witnessed a 13% decline in crime rate in 2023, which was highly praised by politicians and media. However, residents in the US had different feelings regarding the crime rate. A poll in December 2023 found that 77% of Americans believed the crime rate was worsening. The divergence illustrated that the crime rate was still a major concern and a popular topic for politicians and the American people. Therefore, it is meaningful and worthwhile to analyze the crime data to answer the following questions.

First, we aimed to find the most appropriate model to predict the violent crime rate of a county. Our hypothesis for this is that the LightGBM model will have the best performance in prediction compared to other common regression models including linear regression, decision tree, random forest and support vector machine regression, because it can handle large datasets with considerable speed efficiency and can handle the high dimensionality of the data without extensive parameter tuning.

Second, does the number of police officers exhibit a high correlation with the violent crime rate? According to the Guardian News [2], it has been reported that politicians usually attribute the decrease in crime rate to the increase in police staff which we can consider as our hypothesis.

Third, can the age and gender structure of the population explain the high violent crime rate? Criminologists have long recognized that both age and gender are very robust predictors of crime rates [3]. Therefore, we will regard it as our hypothesis and test it with further analysis.

2 Exploratory Descriptive Analysis

2.1 Dataset Overview

The data we analyzed is sourced from a 2023 Kaggle dataset [4] that aggregated data from various government websites and research platforms. This dataset provides insights into violent crime occurrences in U.S. counties, focusing on potential contributing factors. It includes around 2525 entries and covers 117 variables, primarily describing socioeconomic aspects of U.S. counties such as demography, economy, education, housing, policy, diversity, and technology, as shown in Table 1.

Table 1: Variables Summary by Type and from Socio-economic Perspectives

Variable Type	**Variables**
Numerical	**Response:** Violent Crime Rate **Unique Identifier:** FIPS **Demography:** POPESTIMATE, POPEST_MALE, POPEST_FEM, UNDERX_TOT, UNDERX_MALE, UNDERX_FEM, AGEXY_TOT, AGEXY_MALE, AGEXY_FEM, AGEXPLUS_TOT, AGEXPLUS_MALE, AGEXPLUS_FEM, MEDIAN_AGE_TOT, MEDIAN_AGE_MALE, MEDIAN_AGE_FEM **Economy:** PerCapitalInc, PovertyAllAgesPct, NumUnemployed **Education:** Ed1LessThanHSPct, Ed3SomeCollegePct, Ed5CollegePlusPct, **Housing:** OwnHomePct, Housing_Units **Public Safety Workforce:** male_officer, female_officer **Politics:** Republican, Democrat **Diversity:** Diversity-Index, Immigration_Rate_2000_2010, Black or African American alone percentage, Asian alone percentage, American Indian and Alaska Native alone percentage, Native Hawaiian and Other Pacific Islander alone percentage, Two or More Races percentage, Hispanic or Latino percentage, White alone percentage
Ordinal	**Technology:** Tier_1($\geq$200Kbps), Tier_2($\geq$10 Mbps), Tier_3($\geq$25 Mbps), Tier_4($\geq$100 Mbps) **Coding Connection:** - 0: Not available - 1: $0 < X \leq 200$ - 2: $200 < X \leq 400$ - 3: $400 < X \leq 600$ - 4: $600 < X \leq 800$ - 5: $800 < X$

Violent Crime Rate is our numerical response in this project, representing the number of violent crime occurrences per 100,000 population. FIPS (Federal Information Processing Standards Codes) are unique identifiers for US counties.

A majority of float-type variables are demographic, which separates the impacts on criminal occurrences by the different combinations of age and gender in the population groups. The following examples can show how to understand the demographic variables: POPESTIMATE is the total population and POPEST_MALE is the male population. UNDER5_TOT means the total population that is younger than 5 years old. AGE1014_FEM means the female population whose age lies inclusively between 10 and 14. MEDIAN_AGE_MALE is the median age of the male population.

Since there are over 90 age-group variables, we rearranged multiple age-group variables and calculated their percentages of the total population in each county into four variables:

AGE013_NEW_pct, AGE1439_NEW_pct, AGE4059_NEW_pct and AGE60PLUS_NEW_pct only for the convenience of exploratory descriptive analysis (EDA) purposes. For example, We created AGE013_NEW_pct by adding UNDER5_TOT and AGE513_TOT and then dividing this by POPESTIMATE. Defining these new variables ensures a comparable analysis at the EDA stage. For the same reason, we also created MALE_pct, and Housing_Units_avg obtained by using POPESTIMATE to divide POPEST_MALE, and Housing_Units, respectively. In particular, we also created Unemployment_rate by dividing NumUnemployed by the summation of AGE1439_NEW, AGE4059_NEW, and AGE6064_TOT.

Our dataset employs per capita income to measure economic effects and calculates the poverty percentage of the total population. It also tracks the percentage of the unemployed population which is the Unemployment_rate that we just mentioned.

Several percentage-rate variables detail the education level of the population in each county. Ed1LessThanHSPct, Ed3SomeCollegePct, Ed3CollegePlusPct represent the percentages of the population with education less than high school, whose highest level of education is college, who received education beyond college.

OwnHomePct variable records the percentage of the population who own at least one house and Housing Units features the number of available housing for living in a specific US county.

Male_officer and female_officer measure the numbers of male and female police officers in the county. Republican and Democrat respectively represent the percentages of the population who support republicanism or democracy.

For diversity-related variables, Diversity-Index is calculated by the Simpson's diversity index, representing the probability that two randomly selected people are from different races. A higher index of a county means that it has more diversity. In addition, this dataset also includes the average immigration rate from 2000 to 2010. All the other variables represent the percentages of the specified races in a county.

There are only four ordinal variables representing the distribution of internet speeds across households within each county. For example, Tier_1 variable features the number of households with a download speed of at least 200 Kbps by categorical values. If Tier_1 = 0, no one in this county has the internet with a download speed of at least 200 Kbps. Similarly, If Tier_1 = 1, the number of households that have the internet with a download speed of at least 200 Kbps falls within the range of (0, 200] per 1,000 Housing Units.

2.2 Data Diagnostics

This data is split into training and testing subsets at an 80-20 ratio beforehand, based on which we conducted some data diagnostics as follows.

2.2.1 Missing Values Detection

In this dataset, 325 records contain missing values. Upon reviewing these missing values, it became apparent that certain counties were missing numerous variables, likely due to incomplete data collection. Consequently, we decided to remove these 325 records.

2.2.2 Descriptive Analysis

Table 2: Statistics Summary

	count	mean	std	min	25%	50%	75%	max
violentCrimeRate	2200.00	53.70	221.92	0.00	5.00	18.00	47.00	8854.00
POPESTIMATE	2200.00	87137.07	261177.02	51.00	11093.25	25525.00	64087.00	5109292.00
MALE_pct	2200.00	0.50	0.02	0.45	0.49	0.50	0.51	0.78
AGE013_NEW_pct	2200.00	0.17	0.03	0.04	0.15	0.17	0.18	0.28
AGE1439_NEW_pct	2200.00	0.32	0.05	0.16	0.29	0.31	0.34	0.67
AGE4059_NEW_pct	2200.00	0.24	0.02	0.09	0.23	0.24	0.26	0.34
AGE60PLUS_NEW_pct	2200.00	0.27	0.06	0.08	0.24	0.27	0.30	0.53
MEDIAN_AGE_TOT	2200.00	41.67	5.11	20.90	38.80	41.50	44.30	61.70
PerCapitaInc	2200.00	26798.58	5832.54	12903.00	22820.75	26208.50	29814.25	56180.00
PovertyAllAgesPct	2200.00	14.97	5.72	2.60	10.88	14.10	18.00	44.30
Immigration_Rate_2000_2010	2200.00	1.14	1.65	-2.49	0.18	0.55	1.42	17.68
Ed1LessThanHSPct	2200.00	13.45	6.34	1.67	8.74	12.25	17.02	66.34
Ed3SomeCollegePct	2200.00	21.98	3.66	4.12	19.52	21.85	24.18	38.67
Ed5CollegePlusPct	2200.00	21.10	8.79	0.00	14.97	19.08	24.80	63.06
male_officer	2200.00	95.81	248.16	0.00	12.00	32.00	79.00	4529.00
female_officer	2200.00	14.59	53.85	0.00	1.00	3.00	9.00	987.00
OwnHomePct	2200.00	71.98	7.32	31.47	68.16	72.73	77.06	89.66
Unemployment_rate	2200.00	0.03	0.01	0.01	0.02	0.03	0.03	0.13
Housing_Units_avg	2200.00	0.48	0.12	0.22	0.41	0.46	0.53	1.80
Republican	2200.00	0.42	0.05	0.28	0.39	0.42	0.46	0.57
Democrat	2200.00	0.40	0.05	0.25	0.38	0.40	0.43	0.57
Diversity-Index	2200.00	0.29	0.17	0.02	0.12	0.25	0.45	0.74
Black or African American alone percentage	2200.00	8.12	13.01	0.00	0.70	2.10	8.70	84.70
American Indian and Alaska Native alone percentage	2200.00	1.91	5.96	0.00	0.30	0.60	1.20	84.50
Asian alone percentage	2200.00	1.09	1.64	0.00	0.40	0.60	1.10	23.50
Native Hawaiian and Other Pacific Islander percentage	2200.00	0.08	0.14	0.00	0.00	0.00	0.10	2.40
Two or More Races percentage	2200.00	1.77	1.08	0.00	1.20	1.50	2.00	10.30
Hispanic or Latino percentage	2200.00	8.63	13.20	0.40	2.00	3.60	8.62	95.70
White alone percentage	2200.00	79.29	18.17	3.20	69.00	86.00	93.60	98.90

In the summary statistics for numerical variables (Table 2), we find that our response violentCrimeRate has a mean value of approximately 53.70 per 100,000 population, with a very high standard deviation, suggesting that there is significant variation in violent crime rates across different counties. The minimum rate is 0, indicating some counties have no reported violent crimes, while the maximum rate is extremely high at 8854, indicating some counties with a very high number of violent crime occurrences. Several age groups (AGE013_NEW_pct, AGE1439_NEW_pct, etc.), which are demographic variables represented in percentage terms, suggest a relatively even distribution across the age ranges, with the 14-39 age group occupying the largest percentage (mean of 32.00%) of the population on average.

Moreover, the following variables in this dataset have the potential to impact the response based on the findings from their summary of statistics. The average perCapitalInc is 26,798.58 USD, with a standard deviation of 5,832.54 USD, indicating variability but not as pronounced as for violent crime or population estimates. For home ownership, 72% of the population on average are homeowners, with a range from 31% to over 90%, indicating a wide variation in homeownership rates. The average unemployment_rate is quite low around

3%, but it can go as high as around 13%, showing that unemployment is more concentrated in certain areas. Racial Demographics, like the percentages for Black or African American alone, American Indian and Alaska Native alone, and other racial demographics, all have wide ranges depending on the county's racial composition.

2.2.3 Outliers Detection

Figure 1 shows the distribution of the response variable Violent Crime Rate, noting that there are several outliers that might have an impact on the model performance. We detected 29 outliers in total using the z-score method, 14 of which are in the training set.

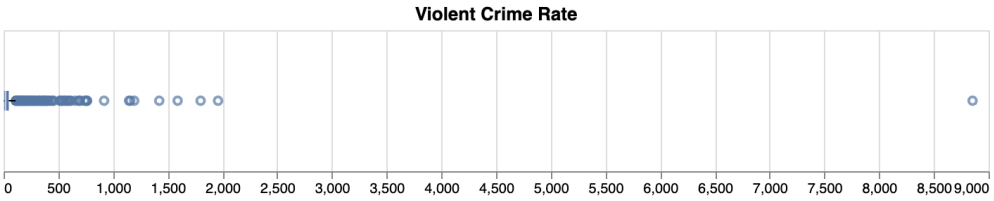


Figure 1: The Distribution of Violent Crime Rate

For better visualization, we limit the values of the response to below 400, accounting for 98% of the responses as shown in Figure 2.

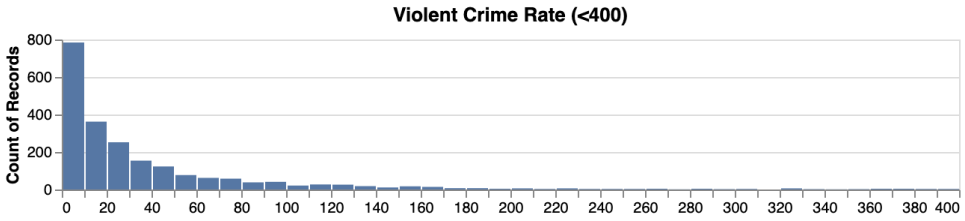


Figure 2: The Distribution of Violent Crime Rate (< 400)

2.2.4 State Mapping

To add geographical variables in the following modeling and analysis, we performed a mapping procedure to categorize counties into a state variable based on the FIPS variable and transformed this state variable into a dummy matrix using one-hot encoding, assigning values of 0 or 1 accordingly.

2.2.5 Multicollinearity

Due to the strong linear correlation between the total population, female population, and male population, we decided to drop these variables to avoid multicollinearity:

- total population in all age groups (UNDERX_TOT, AGEXY_TOT, AGEXPLUS_TOT)

- total population (POPESTIMATE)
- total female population(POPEST_FEM)
- total male population(POPEST_MALE)

2.2.6 Correlation Analysis

To further explore the data, we performed a basic correlation analysis between the violent crime rate and variables related to different socio-economic factors as shown in Figure 3, which revealed three key variables highly correlated with the violent crime rate: POPESTIMATE (correlation coefficient of 0.55), male_officer (correlation coefficient of 0.51), and female_officer (correlation coefficient of 0.49). These strong correlations suggest the following insights.

Firstly, a larger population, as indicated by the POPESTIMATE, appears to correspond to an increase in the crime rate. This aligns with the notion that higher population densities often correlate with higher crime rates due to various socioeconomic and demographic factors.

Secondly, there seems to be a positive correlation between the number of law enforcement officers, both male and female and the local violent crime rate. This correlation suggests that areas with a greater presence of law enforcement officers tend to exhibit higher violent crime rates. However, it is important to note that this correlation does not imply causation; rather, it could be indicative of law enforcement deployment in response to existing violent crime rates rather than being the direct cause of the violent crime rate itself.

Other variables have weaker correlations (below 0.3) to the violent crime rate, indicating that these factors may have less direct influence on the prevalence of violent crimes.

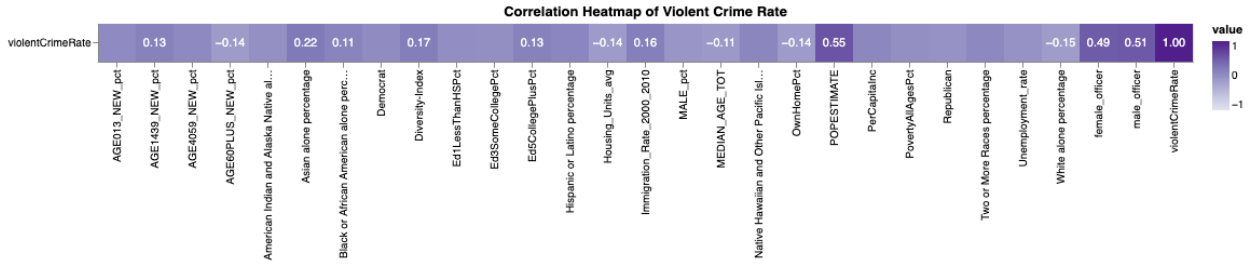


Figure 3: Correlation Heat Map of Violent Crime Rate

3 Methodology

We selected six different methods to test our scientific hypotheses. This section will explain why we selected these methods, how they work respectively, other techniques we applied, and performance evaluation metrics for this analysis. For each method, we performed cross-validation on the data to mitigate over-fitting and used grid search to conduct hyperparameter tuning for the optimal model construction.

After obtaining the best hyperparameters for all methods, we will evaluate and compare their performance and provide the sorted coefficients or feature importance and respective interpretations in the next section.

3.1 Hyperparameter Tuning Techniques

Grid search is a technique used for hyperparameter tuning, where you specify a grid of hyperparameter values to search over different combinations and decide the best hyperparameters based on the results of evaluation metrics (RMSE, R^2) across different cross-validation sets. Cross-Validation (CV) is a technique used to assess how well a model generalizes to new data by splitting the available data into multiple subsets (folds). In the following steps, we chose a 5-fold CV grid search to conduct the hyperparameter tuning.

3.2 Evaluation Metrics

Root Mean Squared Error (RMSE) measures the square root of the average squared difference between the predicted values and the actual values. We can use RMSE to measure the overall accuracy of different models' predictions, with lower values indicating better accuracy.

R^2 measures how well the regression model fits the actual data. It varies between 0 and 1, where higher values signify a stronger fit. The standardized range of R^2 allows us for easier comparison between models, as it provides a uniform basis for evaluating their performance.

3.3 Lasso in Linear Regression

The first model we built is a linear regression with LASSO regularization technique applied, which can filter out the less important features by shrinking the feature coefficients to 0. LASSO is suitable for high-dimensional datasets and feature selection [5]. Therefore, we consider LASSO as an appropriate method for linear regression modeling in our analysis.

The first step we took was standardizing the features to exclude the influence of the scale on LASSO coefficients. Secondly, we detected the outliers by the z-score method and dropped them because LASSO is sensitive to outliers. Then, we tried to find the optimal alpha (the penalty factor ranging between 0.01 and 1, the higher it is, the faster the coefficients will be forced to 0) which could guarantee the convergence of the result within 3000 iterations.

3.4 Decision Tree

Decision tree is a tree-like model that splits the data by one feature at each node and the leaf of the tree is the result of the model. It has no statistical assumptions required compared with the linear regression model and does not require feature scaling, making it easy to use with our dataset that has features in different scales. Moreover, tree-like results exhibit high interpretability in answering scientific questions. However, this method allows for growing deep trees without constraints, leading to poor generalization of unseen data [6].

To avoid an overly deep decision tree (too many nodes and leaves) which could lead to over-fitting of the training data, we tried to find the best decision tree model by tuning the following hyperparameters: the maximum depth (None to 50), the minimum number of samples required at a leaf node (10 to 100), the minimum number of samples required to split an internal node (0 to 50), the maximum number of features allowed at each node.

3.5 Random Forest

The random forest model is an ensemble of decision trees renowned for its ability to capture complex relationships and handle large datasets effectively. Unlike single decision trees, random forest leverages the collective predictions of multiple trees to make more accurate predictions. This ensemble approach not only enhances prediction accuracy but also helps mitigate overfitting, making the random forest model a robust and versatile technique and suitable for our analysis [7].

We fine-tuned several key parameters of the random forest model to achieve the best performance. Parameters included the number of estimators (ranging from 300 to 1000), the number of features allowed at each node (ranging from 10 to 50), the maximum depth of a tree (ranging from 5 to 30), and the minimum samples required at each leaf node (ranging from 5 to 100).

3.6 Light Gradient Boosting Machine

Light Gradient Boosting Machine (LightGBM) is an efficient gradient boosting framework using tree-based algorithms to grow an ensemble of decision trees in a sequential manner. It learns every single tree to correct the error term of the previous trees until the gradient of the loss function reaches a predetermined level or the maximum iterations are reached. Instead of focusing on the entire dataset at every step like other gradient boosting models, LightGBM focuses on larger errors which are the instances it learns from the most, and reduces the dimension of data to improve efficiency and speed [8]. Therefore, LightGBM is particularly suitable for our large, comprehensive, multi-feature dataset even though it still has the potential to cause overfitting, similar to traditional decision tree models.

We fine-tuned the following key hyperparameters of LightGBM: the learning rate (ranging from 0.01 to 0.1, which controls the speed of learning from the data), the number of iterations (ranging from 100 to 500), the percentage of features being used (0.1 to 1.0), and the number of leaves in each tree (ranging from 10 to 100).

3.7 Support Vector Machine Regression

Support Vector Machine Regression (SVR) aims to find the hyperplane that best fits the training data while allowing some tolerance for errors and maximizing the margin to improve generalization to unseen data [9]. When dealing with outliers and high-dimensional data, it can lead to a more accurate and robust model. Since our dataset has over 100 features and also has some outliers in response, the SVR model can be an appropriate choice.

Firstly we standardized the features to avoid the impact of the scale of data on the modeling process. Then, we fine-tuned the hyperparameters of the SVR model including kernel type (linear, radial basis function, or polynomial), C (penalty parameter, ranging from 0.1 to 1000), epsilon(the numerical degree of the tolerance, ranging from 0.1 to 1).

4 Results and Discussion

The R^2 and RMSE of each model are calculated on the testing set and the results are shown in Table 3.

Table 3: Results of Model Performance

	LASSO	Decision Tree	Random Forest	SVR	LightGBM
R^2	0.31	0.34	0.41	0.44	0.44
RMSE	138.43	135.38	127.95	124.43	125.15

The LASSO linear model and decision tree just had an R^2 of around 30% and an RMSE over 135, both of which are inferior to the other models. Random forest, SVR and LightGBM have fairly close R^2 and RMSE, suggesting that all these three models have a relatively better performance in analyzing our data set in comparison with the other two.

In the comparison of the last three models, we consider LightGBM to be the preferred choice over SVR and random forest for our prediction task due to several key advantages. Firstly, LightGBM is good at handling high-dimensional datasets characterized by complex, non-linear relationships between features and the response. Secondly, it can capture interactions between features, making it particularly effective with intricate feature dependencies. Thirdly, its support for high-dimensional data and its robustness against outliers allows it to process large datasets and extreme values efficiently. Therefore, LightGBM is the most appropriate model in terms of the model performance and the suitability to verify our hypotheses. However, due to the nature of the tree-based algorithm, we are unable to obtain the specific directions of the relationship between our response and the features, thus we will extract the coefficient results from LASSO which are consistent with the ones in LightGBM as support to answer our scientific questions.

We listed the most important 20 features suggested by LightGBM and the results are shown in Figure 4.

Based on the feature importance in LightGBM, we can find that diversity-related features, particularly the Immigration Rate, have strong predictive power for our response. Meanwhile, the diversity index in the LASSO linear model exhibits a positive relationship with the violent crime rate, supporting the finding that the more diverse a US county is, the more violent crimes may occur in that county. In addition, LightGBM considers the percentage of Asian, Hispanic or Latino, and Black as important factors for violent crime occurrences. According to the coefficients in the LASSO linear model in Table 4, the percentage of Asians and Hispanic or Latino has a negative impact while the percentage of Black has a positive impact on violent crime.

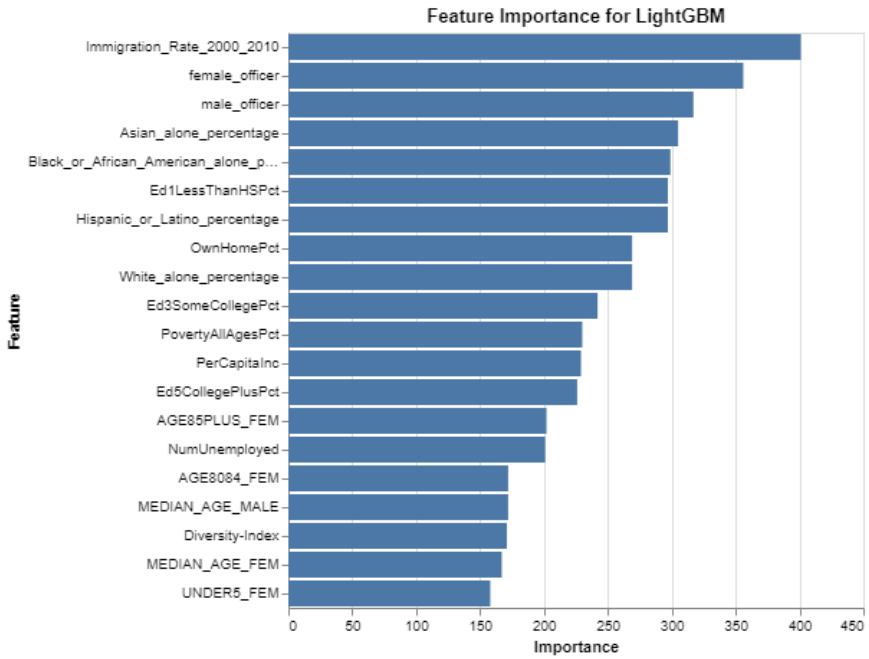


Figure 4: Feature Importance from LightGBM

Table 4: Important Feature Coefficients in LASSO Linear Model

Feature	**Coefficient**
Black or African American alone percentage	8.89
male_officer	8.29
Ed1LessThanHSPct	6.38
Asian alone percentage	-5.25
Diversity-Index	5.16
Hispanic or Latino percentage	-4.61
PerCapitaInc	-4.09

From the perspective of the public safety workforce, the number of officers in a county has a considerable influence on the county's crime rate, which means that violent crime is closely related to the number of officers. However, the results in the LASSO linear model suggest that the number of male officers has a positive relationship with violent crime, which confirms our previous opinions in correlation analysis that they might have a casual relationship implying a county with high occurrences of violent crime has more male officers.

The education level plays an influential role in predicting the number of violent crimes not only in LightGBM but also in the LASSO linear model. In particular, the percentage of the population with an educational level less than high school has a positive impact on our response, which implies people who are less educated are more likely to commit violent crimes. Economic factors also affect the number of violent crimes in the US counties. Income per capita plays a crucial role in the prediction of violent crime as shown in the coefficients in

the LASSO linear model, suggesting higher income may potentially decrease the occurrences of violent crime in a county.

5 Conclusions and Recommendations

5.1 Conclusions

In conclusion, LightGBM is the most appropriate model among all the choices for predicting the US county violent crime rate in terms of performance and interpretability, which is in line with our first hypothesis.

Based on our findings in the last section, the number of police officers does have a significant influence on the violent crime rate, which is in line with our second hypothesis. However, as mentioned before, it could be indicative of law enforcement deployment in response to existing crime rates rather than being the direct cause of the crime rate itself.

The age and gender structures of the population do not exhibit a significant relationship with the violent crime rate, which is against our third hypothesis. However, the ethnic structure of the population has a strong influence on the violent crime rate.

5.2 Recommendations

Based on the conclusions above, we recommend using LightGBM to predict and select counties with low crime rates.

The public safety workforce has a significant influence on the crime rate and should be considered by residents and policymakers. Residents might be cautious with counties that have many male police officers when selecting areas with low violent crime rates. For policymakers, we recommend prioritizing investments in law enforcement resources, including recruitment, training, and community policing strategies, but understanding the exact influence of the public safety workforce on the violent crime rate needs further analysis.

There is no significant relationship between the age and gender structure of the population and the violent crime rate, so residents and policymakers can pay less attention to these kinds of features when making decisions regarding the county's crime rate. In terms of predicting and understanding the violent crime rate, we suggest both residents and policymakers focus more on the ethnic structure.

5.3 Future Analysis

To better understand the impact of different factors and enhance our analysis of the violent crime rate, we can potentially consider the following aspects:

Quantifying feature influence: we can delve deeper into the numerical relationship between the features selected from LightGBM and the crime rate. This analysis will enable us to precisely quantify the influence of each feature on the crime rate, providing valuable insights into which factors have the most significant impact.

Incorporating geometric features: we can exploit additional features, such as geometric features, in our analysis. Incorporating geometric data can provide a more comprehensive understanding of crime dynamics and contribute to more precise predictions.

Analyzing at a granular level: Given the substantial variation in crime dynamics across smaller geographic areas, we should consider conducting analyses and building models at the city or community level. Understanding crime trends at a granular level can facilitate targeted interventions and policy decisions tailored to specific communities.

References

[1] K. Dilanian. (2024, March) The u.s. crime rate is still dropping. Accessed: 2024-03-23. [Online]. Available: https://www.nbcnews.com/news/crime-courts/us-crime-rate-still-dropping-says-fbi-rcna144100

[2] A. Gabbatt. (2024) Fbi data shows us crime plummeted in 2023 but experts warn report is incomplete. Accessed: 2024-03-23. [Online]. Available: https://www.theguardian.com/us-news/2024/mar/19/fbi-data-shows-us-crime-plummeted-2023

[3] J. F. Sheley, "Gender, age, and crime," *Criminology*, pp. 67–93, 1991.

[4] L. Liu, "Number of violent crimes by county," 2024, accessed: 2024-03-23. [Online]. Available: https://www.kaggle.com/datasets/leoliu1415/number-of-violent-crimes-by-county

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, Jan 1996.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer Series in Statistics, 2009.

[7] V. Svetnik, A. Liaw, C. Tong, J. Culberson, R. Sheridan, and B. Feuston, "Random forest: A classification and regression tool for compound classification and qsar modeling," *Journal of Chemical Information and Modeling*, vol. 43, no. 6, pp. 1947–1958, 2003.

[8] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[9] S. Gunn, "Support vector machines for classification and regression," ISIS, Technical Report, May 1998.

Appendix A Code

Data Preprocessing

In [40]:
```
# import the package and read the data
import pandas as pd
import numpy as np
import seaborn as sns
import altair as alt
import missingno as msno
import lightgbm as lgb
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.pipeline import Pipeline
crime_train = pd.read_csv("/content/train.csv",index_col=0)
crime_test = pd.read_csv("/content/test.csv", index_col=0)
crime_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1975 entries, 0 to 2518
Columns: 117 entries, FIPS to Violent crime
dtypes: float64(115), int64(2)
memory usage: 1.8 MB
```

In [24]:
```
pd.set_option('display.max_columns', None)
def findRecordsWithNAs(numOfNA):
  columns_with_missing =crime_train.columns[crime_train.isnull().sum() == numOfNA]
  filtered_columns = crime_train[columns_with_missing]
  return crime_train[filtered_columns.isnull().any(axis=1)]

list = []
for i in [4, 9, 14, 312]:
  list.append(findRecordsWithNAs(i).FIPS.to_list())

flat_list = [item for sublist in list for item in sublist]


has_duplicates = len(flat_list) != len(set(flat_list))
print(has_duplicates) # there is duplicates
print(len(flat_list))
print(len(set(flat_list)))
FIPSToDrop = set(flat_list)
# 325 out of 1975 counties records to be dropped
```

```
True
339
325
```

In [25]:
```
# Drop Missing Values
pd.set_option('display.max_rows', None)
```

1/10

```
cleaned_crime_train = crime_train[~crime_train['FIPS'].isin(FIPSToDrop)]
from sklearn.model_selection import train_test_split
cleaned_crime_train[cleaned_crime_train.isnull().any(axis=1)]
```

Out[25]:

	FIPS	POPESTIMATE	POPEST_MALE	POPEST_FEM	UNDER5_TOT	UNDER5_MALE	UNDER

In [26]:
```
crime_test.isnull().sum() #no missing values in test set
crime_test[crime_test.isnull().any(axis=1)]
```

Out[26]:

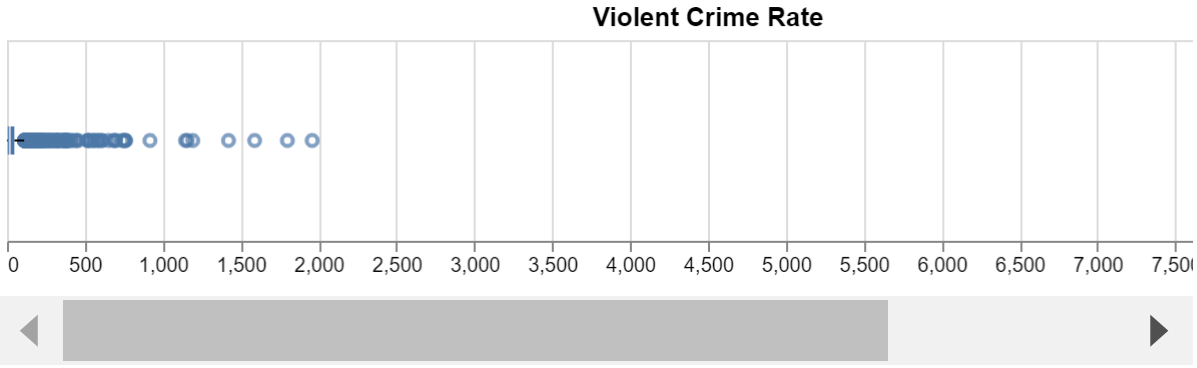
	FIPS	POPESTIMATE	POPEST_MALE	POPEST_FEM	UNDER5_TOT	UNDER5_MALE	UNDER

In [27]:
```
# Since there is no missing values in test set, merge test set with the cleaned tra
#cleaned_full_dataset = pd.concat([cleaned_crime_train, crime_test], ignore_index=T
#print(cleaned_full_dataset.isna().sum())
new_train_set = cleaned_crime_train
new_test_set = crime_test
new_train_set.rename(columns={'Violent crime': 'violentCrimeRate'}, inplace=True)
new_test_set.rename(columns={'Violent crime': 'violentCrimeRate'}, inplace=True)
#cleaned_full_dataset.rename(columns={'Violent crime': 'violentCrimeRate'}, inplace
```

```
<ipython-input-27-2b68995711f7>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  new_train_set.rename(columns={'Violent crime': 'violentCrimeRate'}, inplace=True)
```

In [28]:
```
# plot the distribution plot for response value to detect outliers
df=pd.concat([new_train_set, new_test_set], ignore_index=True)
violentCrimeRate_box=alt.Chart(df).mark_boxplot().encode(
    x=alt.X('violentCrimeRate',title=None),
    tooltip = [alt.Tooltip('violentCrimeRate', title='violent Crime Rate')]
    ).properties(height=100,width=700,title='Violent Crime Rate')
violentCrimeRate_box
```

Out[28]:



In [29]:

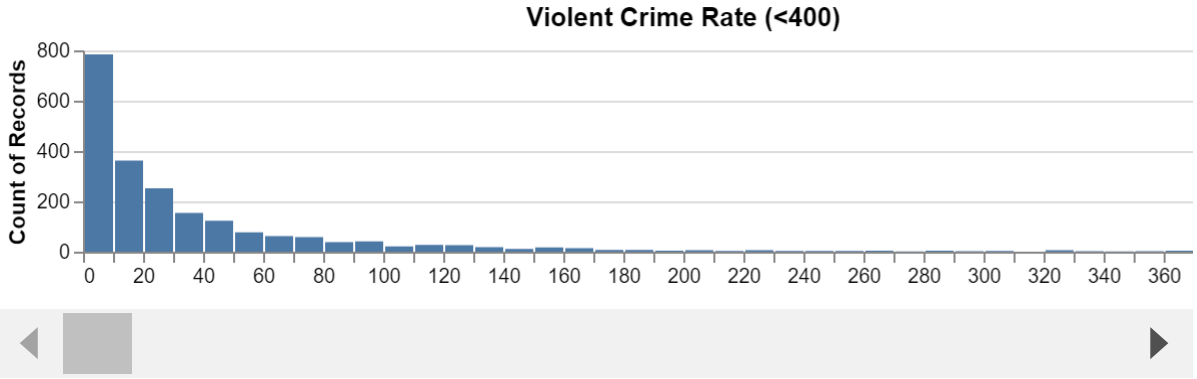
```
# calculate the number of outliers using z-score method
z_scores = np.abs((df['violentCrimeRate'] - df['violentCrimeRate'].mean()) / df['vi
threshold = 2
outliers = df[z_scores > threshold]
print(len(outliers))
print(round(len(df[df['violentCrimeRate']<400])/len(df), 2))
```

```
29
0.98
```

In [9]:

```
# plot barchart of response values less than 400
violentCrimeRate_bar=alt.Chart(df).mark_bar().encode(
    x=alt.X('violentCrimeRate',bin=alt.Bin(maxbins=1000),
            title=None,scale=alt.Scale(domain=[0, 400])),y=alt.Y('count()')
            ).properties(height=100,width=600,title='Violent Crime Rate (<400)')
violentCrimeRate_bar
```

Out[9]:



In [30]:

```
def drop_tot_columns(df):
    """
    Drops all columns from the DataFrame that end with '_TOT'.

    Parameters:
    - df: pandas.DataFrame

    Returns:
    - pandas.DataFrame without columns ending with '_TOT'
    """
    # Use the filter method to find columns that end with '_TOT'
    cols_to_drop = df.filter(like='_TOT').columns

    # Drop these columns from the DataFrame
    df_dropped = df.drop(columns=cols_to_drop)
```

```
    return df_dropped

new_train_set = drop_tot_columns(new_train_set)
new_test_set = drop_tot_columns(new_test_set)

new_train_set.drop(columns=["POPESTIMATE", "POPEST_MALE", "POPEST_FEM"], inplace=Tr
new_test_set.drop(columns=["POPESTIMATE", "POPEST_MALE", "POPEST_FEM"], inplace=Tru

# Mapping FIPS to State
county_fips_df = pd.read_csv('/content/county_fips.csv')

new_train_set = pd.merge(new_train_set, county_fips_df, left_on='FIPS', right_on='f
new_test_set = pd.merge(new_test_set, county_fips_df, left_on='FIPS', right_on='fip

new_train_set.drop('FIPS', axis=1, inplace=True)
new_test_set.drop('FIPS', axis=1, inplace=True)
encoder_train = OneHotEncoder(sparse_output=False)
encoder_test = OneHotEncoder(sparse_output=False)
state_encoded_train = encoder_train.fit_transform(new_train_set[['state']])
state_encoded_test = encoder_test.fit_transform(new_test_set[['state']])

state_encoded_df_train = pd.DataFrame(state_encoded_train, columns=encoder_train.ge
state_encoded_df_test = pd.DataFrame(state_encoded_test, columns=encoder_test.get_f

new_train_set = pd.concat([new_train_set, state_encoded_df_train], axis=1)
new_test_set = pd.concat([new_test_set, state_encoded_df_test], axis=1)

new_train_set.drop(['name','state','fips'], axis=1, inplace=True)
new_test_set.drop(['name','state','fips'], axis=1, inplace=True)

state_CA_data = [0] * 1650
# Inserting the new column at position 1 (so it becomes the second column)
new_train_set.insert(loc=87, column='state_CA', value=state_CA_data)

df_x_train = new_train_set.drop(['violentCrimeRate'], axis=1)
df_y_train = new_train_set.loc[:, ['violentCrimeRate']]
df_x_test = new_test_set.drop(['violentCrimeRate'], axis=1)
df_y_test = new_test_set.loc[:, ['violentCrimeRate']]
```

LASSO

In [31]:
```
# exclude the outliers
outlier = df_y_train.nlargest(14, columns=['violentCrimeRate']).index
df_x_train_ex = df_x_train.drop(outlier, axis=0)
df_y_train_ex = df_y_train.drop(outlier, axis=0)

# standardize data before building LASSO
df_x_train_standardized = pd.DataFrame(columns=df_x_train_ex.columns)
df_x_test_standardized = pd.DataFrame(columns=df_x_train_ex.columns)
for each in df_x_train.columns:
    if 'state' not in each:
      scalar = StandardScaler().fit(df_x_train_ex.loc[:, [each]])
      df_x_train_standardized[each] = scalar.transform(df_x_train_ex.loc[:, [each]]
      df_x_test_standardized[each] = scalar.transform(df_x_test.loc[:, [each]]).fla
```

```
    else:
      df_x_train_standardized[each] = df_x_train_ex[each]
      df_x_test_standardized[each] = df_x_test[each]
df_x_train_standardized.fillna(0, inplace=True)

# build Lasso model using the whole training data set
lasso_model = Lasso(alpha=0.5, max_iter=1000).fit(df_x_train_standardized, df_y_tra

# report the coefficients of Lasso Model
coefficients = pd.DataFrame(lasso_model.coef_, index=df_x_train_ex.columns.to_list(
print(coefficients[abs(coefficients['Coefficient'])>5])

# calculate the R2 and MSE of Lasso model based on the test data
lasso_r2 = r2_score(df_y_test, lasso_model.predict(df_x_test_standardized))
lasso_rmse = np.sqrt(mean_squared_error(df_y_test, lasso_model.predict(df_x_test_st
print(f'The R2 of LASSO is: {round(lasso_r2, 2)}\nThe RMSE of LASSO is: {round(lass
```

```
                                          Coefficient
AGE2024_MALE                                 6.770879
AGE5559_MALE                               -18.971214
AGE7579_MALE                                92.876808
AGE85PLUS_FEM                              -58.333482
Ed1LessThanHSPct                             6.377904
OwnHomePct                                   5.551640
NumUnemployed                              -13.802224
Tier_4                                      13.088907
male_officer                                 8.294105
Diversity-Index                              5.155254
Black or African American alone percentage   8.889592
Asian alone percentage                      -5.245554
The R2 of LASSO is: 0.31
The RMSE of LASSO is: 138.43
```

Decision Tree

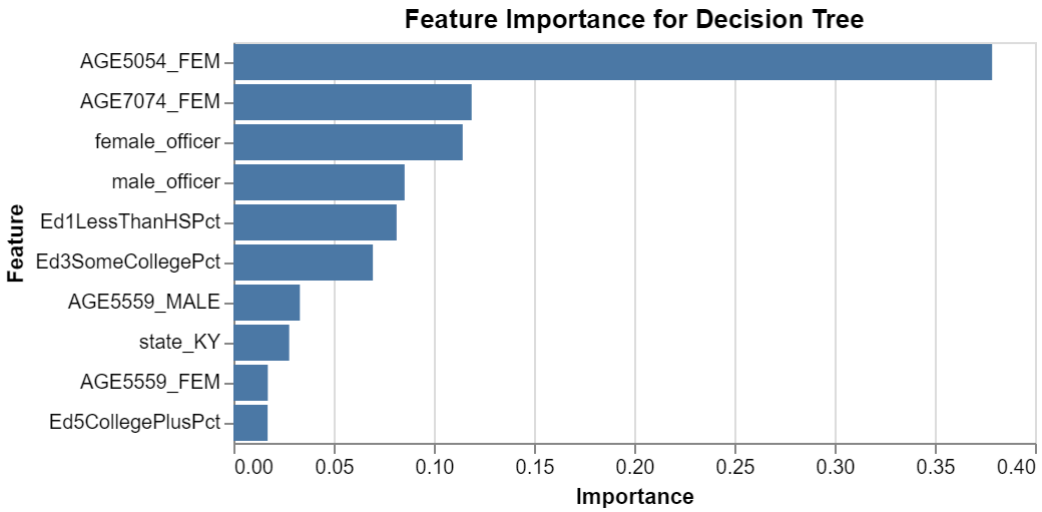
In [32]:
```
# build the decision tree model
dtr_model = DecisionTreeRegressor(criterion='absolute_error', max_depth=None, max_f
dtr_model = dtr_model.fit(df_x_train, df_y_train)
dtr_pred = dtr_model.predict(df_x_test)

# Calculate the evaluation matrics on the test data
dtr_rmse = np.sqrt(mean_squared_error(df_y_test, dtr_pred))
dtr_r2 = r2_score(df_y_test, dtr_pred)

print(f"The R2 of Decision Tree is: {round(dtr_r2, 2)}")
print(f"The RMSE of Decision Tree is: {round(dtr_rmse, 2)}")
```

```
The R2 of Decision Tree is: 0.34
The RMSE of Decision Tree is: 135.38
```

In [33]:
```
dtr_feature = pd.DataFrame({
    'feature': df_x_train.columns.tolist(),
    'importance': dtr_model.feature_importances_})

dtr_feature = dtr_feature.nlargest(10, columns=['importance'])
```

```
alt.Chart(dtr_feature, title='Feature Importance for Decision Tree').mark_bar().enc
    x = alt.X('importance', title='Importance'),
    y = alt.Y('feature', title='Feature', sort='-x')
)
```

Out[33]:



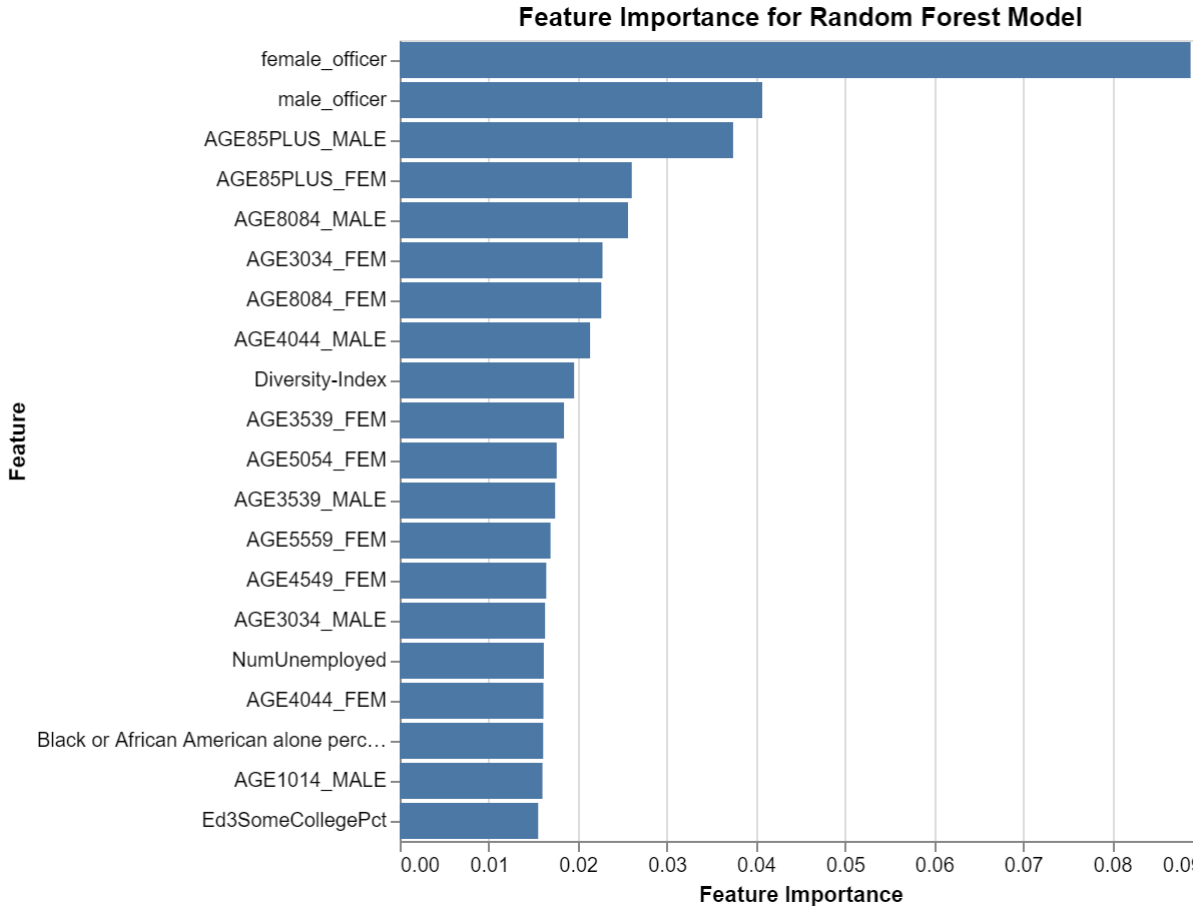
Random Forest

In [34]:

```
rf_model = RandomForestRegressor(max_depth=20, max_features=30, min_samples_leaf=5,
rf_pred = rf_model.predict(df_x_test)
rf_r2 = r2_score(df_y_test, rf_pred)
rf_rmse = np.sqrt(mean_squared_error(df_y_test, rf_pred))
print(f'The R2 of Random Forest is: {round(rf_r2, 2)}\nThe RMSE of Random Forest is

# report the feature importance from random forest model
rf_feature = pd.DataFrame({
    'feature': df_x_train.columns,
    'importance': rf_model.feature_importances_
})

rf_feature = rf_feature.nlargest(20, columns=['importance']).sort_values(by=['impor

alt.Chart(rf_feature, title='Feature Importance for Random Forest Model').mark_bar(
    x = alt.X('importance', title='Feature Importance'),
    y = alt.Y('feature', title='Feature', sort='-x')
)
```

```
The R2 of Random Forest is: 0.41
The RMSE of Random Forest is: 127.95
```

Out[34]:



SVR

In [35]:

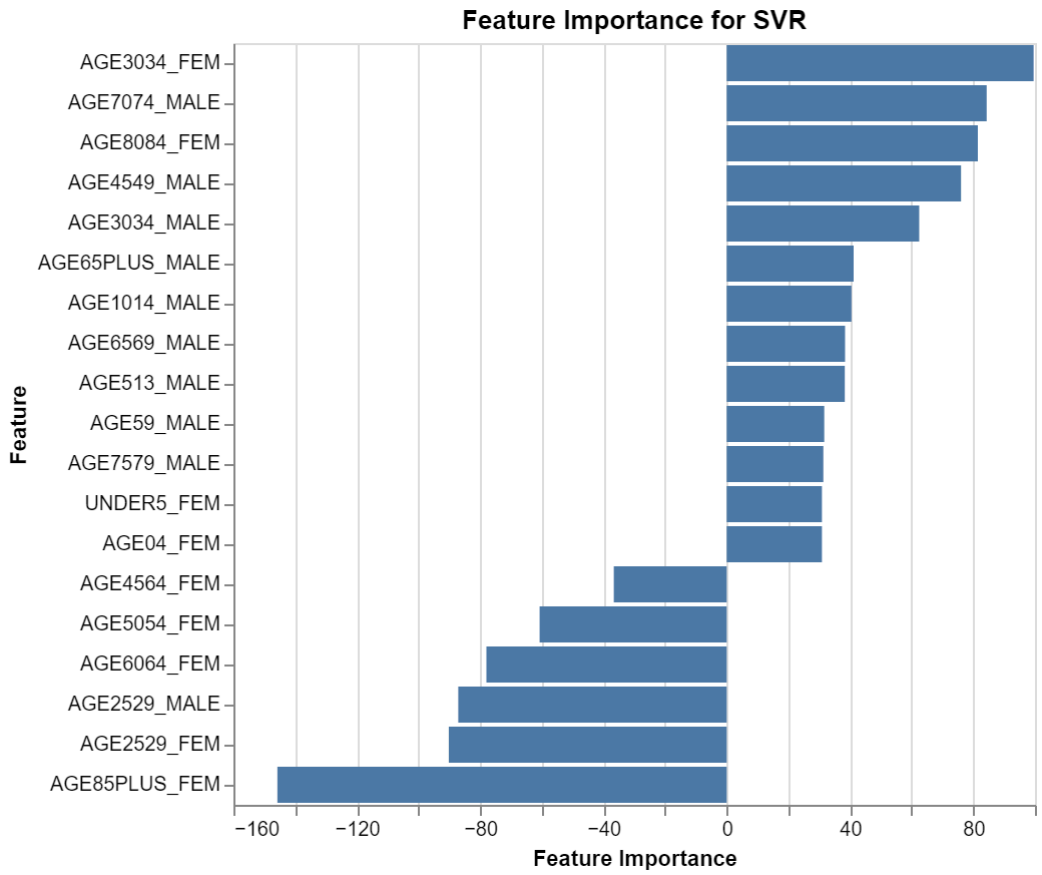
```
# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(df_x_train)
X_test_scaled = scaler.transform(df_x_test)

# Create a pipeline with scaling and SVM regressor
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('svm', SVR(C=100, epsilon=1, kernel='linear'))
])

# Fit the GridSearchCV object to the training data
pipeline.fit(X_train_scaled, df_y_train['violentCrimeRate'])
# Predict the test data
svr_pred = pipeline.predict(X_test_scaled)

# Calculate the evaluation matrics of SVR model
svr_r2 = r2_score(df_y_test, svr_pred)
svr_rmse = np.sqrt(mean_squared_error(df_y_test, svr_pred))
```

```
# Report the result of SVR model
print(f'The R2 of SVR is {round(svr_r2, 2)}\nThe RMSE of SVR is {round(svr_rmse, 2)
```

```
The R2 of SVR is 0.44
The RMSE of SVR is 124.43
```

In [36]:
```
svr_feature = pd.DataFrame({
    'importance': pipeline.named_steps['svm'].coef_[0].astype(float),
    'feature': df_x_train.columns.tolist()
})

svr_feature = svr_feature[abs(svr_feature['importance']) > 30]

alt.Chart(svr_feature, title='Feature Importance for SVR').mark_bar().encode(
    x = alt.X('importance', title='Feature Importance'),
    y = alt.Y('feature', title='Feature', sort='-x')
)
```

Out[36]:



LGBM

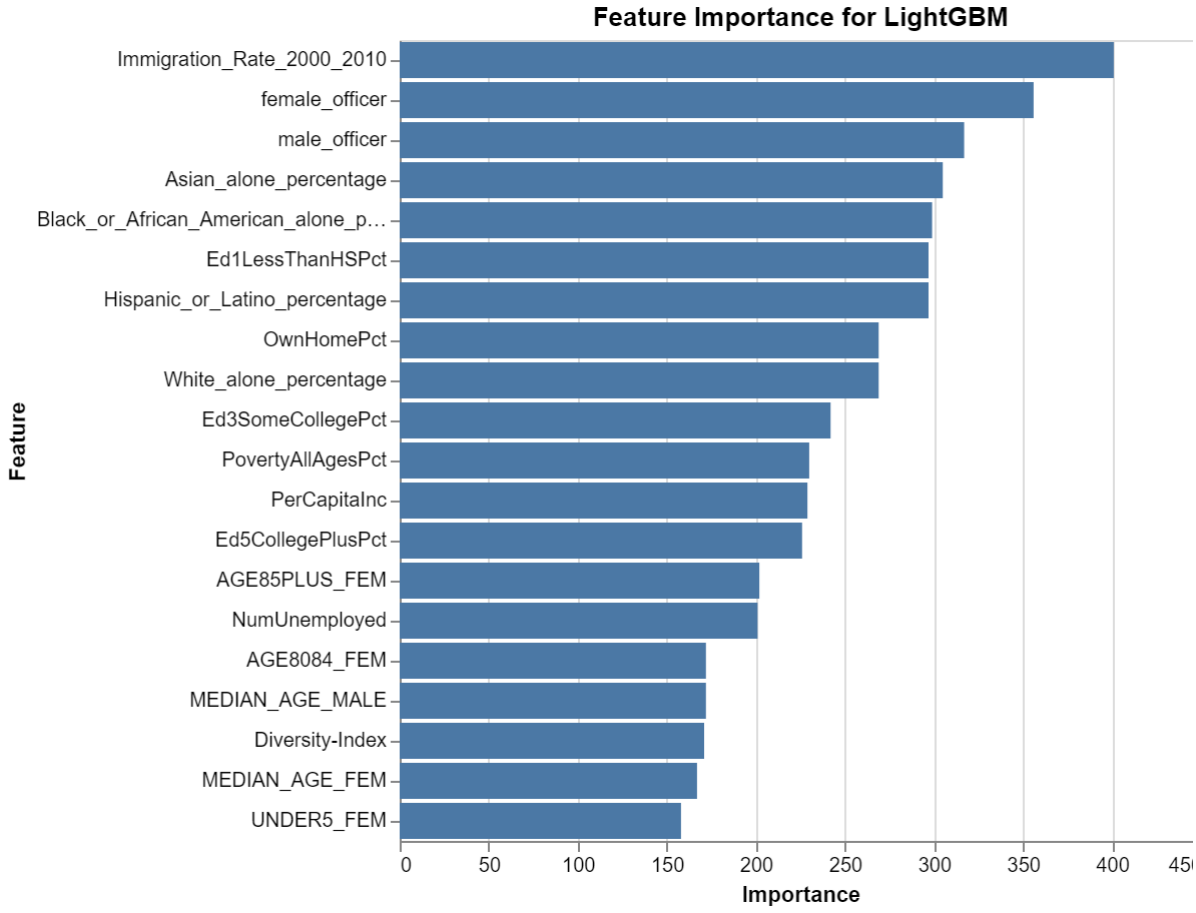
In [37]:
```
lgbm_model = lgb.LGBMRegressor(colsample_bytree=0.8, learning_rate=0.05, n_estimato
# Initialize the Grid Search model


# Fit the grid search to the data
lgbm_model = lgbm_model.fit(df_x_train, df_y_train)

# Make predictions with the best model on the test data
```

```
lgbm_pred = lgbm_model.predict(df_x_test)

# Calculate MSE and R^2 score on the test data
lgbm_rmse = np.sqrt(mean_squared_error(df_y_test, lgbm_pred))
lgbm_r2 = r2_score(df_y_test, lgbm_pred)

print(f"The R2 of GBM is: {round(lgbm_r2, 2)}")
print(f"The RMSE of GBM is: {round(lgbm_rmse, 2)}")
```

```
[LightGBM] [Warning] Found whitespace in feature_names, replace with underlines
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing wa
s 0.001990 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 18690
[LightGBM] [Info] Number of data points in the train set: 1650, number of used featu
res: 113
[LightGBM] [Info] Start training from score 51.336364
The R2 of GBM is: 0.44
The RMSE of GBM is: 125.15
```

In [38]:
```
lgbm_feature = pd.DataFrame({
    'feature': lgbm_model.feature_name_,
    'importance': lgbm_model.feature_importances_
})

lgbm_feature = lgbm_feature.nlargest(20, columns=['importance'])

alt.Chart(lgbm_feature, title='Feature Importance for LightGBM').mark_bar().encode(
    x = alt.X('importance', title='Importance'),
    y = alt.Y('feature', title='Feature', sort='-x')
)
```

Out[38]:



Compare the Model Performance

In [39]:

```
# report the performance matrix of each model
data = {
    'R2': [lasso_r2, dtr_r2, rf_r2, svr_r2, lgbm_r2],
    'RMSE': [lasso_rmse, dtr_rmse, rf_rmse, svr_rmse, lgbm_rmse]
}

# Create a DataFrame with an index that labels each row
performance = pd.DataFrame(data, index=['LASSO', 'Decision Tree', 'Random Forest',

# Display the DataFrame
performance
```

Out[39]:

	LASSO	Decision Tree	Random Forest	SVR	LightGBM
R2	0.31	0.34	0.41	0.44	0.44
RMSE	138.43	135.38	127.95	124.43	125.15