The Crime in California's Cities

For

Comp 541 Data Mining

by

Aigerim Toleukhanova

Tigran Manukyan

05.02.2024.

Data Mining Project Assigned for Comp 541

At

California State University, Northridge

Executive Summary

The project is about exploring and predicting crime, specifically, the number of murders, in California's Cities using data about different crimes committed in said cities from 2005 to 2019. We gathered our data from the FBI web page, each year separately, and then we cleaned and merged the data using well-known data mining techniques. We deployed a web scraper to download the data from the FBI website. Some feature engineering processes used include the reduction of unnecessary/irrelevant columns and the addition of new attributes such as the density of the population. All data before and after cleaning is stored in a GitHub repository.

The most cited data mining prediction algorithms included Linear Regression Rate, Random Forest, Gradient Boosting, and Bayesian Network algorithms. After a literature review of related topics, we deployed Linear Regression and Random Forest algorithms.

We both used 5-fold cross-validation for training and testing the data. RMSE, MSE, MAE, MAPE, and Confusion Matrix were used to evaluate the accuracy of each model. Each algorithm predicted the murder number relatively accurately. Based on the result we figured out which algorithm is better for predicting the murder rate in California.

Table of Content

Data Preparation: Extraction

Data preparation is an essential and crucial part of any data mining process. First, we started by collecting the data we needed for our project. We used data collected by the FBI, which is used for their reports. The data can be found on their website (FBI — Crime in the U.S.). FBI's website allows filtering data by state, and within the state, it has specific tables that sort the data into counties, cities, etc. We were planning to use records from 2005 to 2019 for finding patterns for prediction, therefore we decided to write a scraper, which will allow us to save time, rather than manually collect the data.

We wrote a simple web scraper that traveled through the FBI website and downloaded the table for each year. It jumped from year to year modifying the link for each year. For 2005-2009 it jumps directly to the table. For 2010 onward it goes to the link with the most similar address for the different dates, then goes true the pages until it finds the correct table and downloads it.

After the web scraper did its thing we had 15 different tables, one for each year. Each table has anywhere from 13 to 15 attributes. Some tables had attributes that were combined together in other tables. Each table also had around 460 tuples. Some years had more tuples, some had less, this was because the FBI included some small cities in some years, but didn't include them for other years.

Data Preparation: Data Cleaning/Data Integration/Data Selection

The libraries we utilized for cleaning in R contained: tidyverse, vroom, and readr. All three provide a coherent system for data manipulation, visualization, and modeling.
We started by creating an empty list and downloading all 15 files, then assigning it to that empty list(see Table 2). Each file we downloaded looked like Table 1 Appendix A, where each tuple is a string by default; thus, we had to convert everything into an integer and assign each number for each city for further calculation. By using code samples below:

```
for(i in 4:ncol(allSamples$crimeCalifornia2012)){
  allSamples$crimeCalifornia2012[[i]]<-as.numeric(gsub(",", "",
allSamples$crimeCalifornia2012[[i]]))
}
```

Next, we combined all files into one file named by *combinedAllSamples.* (See Table 3) Where we noticed the redundancy (with a grammatical mistake, using commas between words, etc) in the namings of features. For instance, "Violent" crime is spelled in four different ways: "Violent", "Violentcrime", "Violent..crime", "Violent.crime". We had to merge all those columns into one.
The library tidyverse contains dplyr that helps to manipulate (mutate, filter, etc) the document more easily.

```
combinedAllSamples <- combinedAllSamples %>%
  mutate(Violent = case_when(
    !is.na(Violentcrime) ~ Violentcrime,
    !is.na(Violent.crime) ~ Violent.crime,
    !is.na(Violent..crime) ~ Violent..crime,
    TRUE ~ NA_real_
  ))%>%select(-Violent.crime, -Violent..crime, -Violentcrime)
```

We combined "Rape", "Forcible.rape", "Rape", "Rape.legacydefinition.2", and some of the rape-related features into one "Rape." In such a manner, we mutated all the below columns:

- City
- Violent
- Murder
- Rape
- Assault
- Property Crime
- Theft
- Arson

At last, we named a new dataset "**mergedMerder"**, then proceeded to the next data preparation.

Data Preparation: Dealing with NAs

The Data, after feature engineering, contained multiple NAs per column. To deal with missing data, we plotted the histogram of each column and saw if the distribution was Normal or Skewed. In Normally distributed cases (see Table 5), we assigned each NA with a mean value of the data, and for the Skewed distribution, we assigned the median of the data for missing values (See Table 6). Features such as Assault, Property Crime, Violent, and Burglary had less than 10 tuples missing, thus it made sense to remove the row with unknown rather than using distribution.

My group partner and I decided that area or more specifically density of population per mile would affect the predicting the number of murders. Therefore, we obtained additional data on the square mileage of each city. We called a dataset Area. It contained 1523 unique cities and its square mileage. By default, all tuple structures were characters and looked like the sample below (also, see Table 7):

Los Angeles, CA / 3,862,210 /468.67 sq mi

Therefore, we deleted unnecessary parts of information (such as ", CA / 3,862,210"), left the name of the city, and turned square mileage into a numeric representation for further calculation. Since our mergedMerder data contained only 473 unique cities while the area had 1523, the filtered area dataset was based on the names of the common ones. At this step, we noticed that there was redundancy in the names of the cities we must fix: for example, "Rancho Santa Margarit$", and "Rancho Santa Margarita" were indeed the same cities. After fixing the names of the cities, we merged the datasets based on matching city names.

```
mergedMurder<-merge(mergedMurder, area,  by.x = "NameCity", by.y = "City", all.x = TRUE)
```

The merging process created unknown data: our newly created SQmiles feature had 139 tuples missing (see Table 8). Once more we deployed a histogram to learn about the distribution for finding the missing data. The data on the area was skewed, thus we substituted missing values with a median. We were more interested in the density of the population over the land than the area of the city; thus we created a new feature named "Density" by dividing the population over the area.

Metropolitan cities like Los Angeles, or San Diego were reasons why we had to filter the data (see Table). The cities with such a large population are outliers for our data. We decided to filter data on the population since it affected our data distribution. We used Q1/Q3 filtering, which requires identifying the first/third quantiles of the data and then calculating the range of their difference. Based on that, we calculated the lower and upper bound, then filtered data (3 is a multiplier, we tweaked it based on how we wanted to assign the range):

```
IQR <- Q3 - Q1
lower <- Q1 -3* IQR
 upper <- Q3 + 3 * IQR
```

Data Transformation

We normalized data by using the min-max normalization we covered in the lecture (sample code):

```
minMaxScale<- function(x){ (x-min(x))/(max(x)-min(x))}                    normMergedMurder<-
as.data.frame((lapply(numericMergedMurder, minMaxScale)))
```

Table 10 has samples of the normalized dataset.

Visualization

Visualization is an important process in data observation because it can help you solve potential issues: for example, by deploying the histogram, we predicted/substituted unknowns (NAs) to mean or median; by running a scatter plot(see Table 12) we saw if attributes are correlated, or have any relationship between each other; boxplot(see Table 9 and Table 13) helped us to gain information on quartiles, means, medians of the features. By displaying the data as a graph, we were able to see the data from different angles, meaning we were able to find the relationship between attributes vs target, and attributes vs attributes.

It is an important process for possibly reducing unnecessary, redundant, and unrelated attributes. This will play a crucial role in finding patterns in the future. We created a correlation matrix, that explicitly calculated the correlation between attributes (see Table 11 and Table 12). For example, indices of city or year data collected do not correlate with most of the attributes; thus we will drop CityNumber and Year features for training the model.

Another example is the Theft and Property Crime correlation, which was `0.991722924;` since both are highly correlated with each other, we could drop one of the features as well.

Data Mining/Pattern Evaluation: Linear Regression Analysis (by Aigerim)

In the process of visualization of data, we discovered that attributes are positively correlated and linearly dependent with each other. Especially, the target (label) class is highly correlated (has linear dependency) with multiple feature classes. It was good news since there are known prediction algorithms that lean on the linearity of attributes such as Linear Regression, Lasso Regression, etc.

I decided to use the Linear Regression Algorithm and build my model based on information gathered prior. The R has built-in libraries that helped me to mine the data. "In statistics, **linear regression** is a statistical model which estimates the linear relationship between a scalar response and one or more explanatory variables." (*Linear Regression*. Wikipedia) See Table 16 for the formula of Linear Regression.

The libraries deployed during the data mining process are Metrics, Tidyverse, Dplyr, Caret, Corrplot, and Plyr. Each had a variety of functions, which helped me build, run, and evaluate the model I chose.

For evaluation, I used the k-fold cross-validation metric (from "Caret" and "Corrplot" libraries), where I used multiple k's such as 3, 5, 7, and 10. In addition, I created formulas with a variety of hyperparameters, that I set up before running and training my model. Below is a sample code of the training model:

```
modelLR<-train(formula, data = mergedMurder,method = "lm",
               trControl =train_control)
```

Somewhat 5, 7, and 10-fold cross-validation results were very close in accuracy, thus I will record only 5-fold cross-validation results.

Additionally, using a normalized dataset versus using not normalized dataset made no difference in the final result. Since our data contained over 6,000 observations, my machine had no problem executing it in under 3-5 seconds: the time complexity of the training process was very quick (for the linear regression model).

To evaluate the performance accuracy of the model, we deployed the build in functions, such as:

- Mean Absolute Error - mae
- Mean Square Error - mse
- Root Mean Square Error - rmse
- Mean Absolute Percentile Error - mape
- Confusion Matrix

Calculating most of the above errors needed simply one line of the code. Since Linear Regression Algorithm gets specific values for prediction, the confusion matrix was somewhat inapplicable. The confusion matrix shows how accurately the model classified the solution(in our case prediction). We found the solution for using a confusion matrix: we tweaked our output of real values and predicted value, by assigning it into a certain range. Thus, we turned the regression value prediction problem into a classification and can use the confusion matrix to validate the result.

Data Mining//Pattern Evaluation: Random Forest Analysis (by Tigran)

A Regression Random Forest algorithm was used for prediction. The Random Forest algorithm uses multiple decision trees to predict the outcome. For a classification problem Random Forest would use the most occurring outcome from all the decision trees, for our problem, a regression problem, the average outcome out of all the decision trees is considered as the correct output. K-fold cross-validation method was used for training and evaluation, where K is set to 5. A K-fold cross-validation tends to produce better outcomes than dividing the data set into a training set and a test set, thus we went with the K-fold.

In terms of tuning the Random Forest, 2 variables were considered. First, there was the number of trees constructed. After running a test algorithm on 1000 trees, the data showed that while at first the quality of the outcome depended on the number of trees, eventually the graph flattened out, and there is no significant difference between 500 and 1000 trees (see Table 17). So, the final algorithm is run on 500 trees to help on run time. Next is the number of variables considered at each split, which we will call mtry, which needs to be optimized. For Random Forests, at each split for the Decision Tree a set of random attributes is selected from the input list, and then the best

attribute from the subset is selected for the split point. The Random Forest algorithm ran 10 times changing mtry for each run going from 1 to 10 and comparing the Root-Mean-Square Error, we see that the RMSE is lowest when mtry = 5 (see Table 18). Thus the final Random Forest algorithm uses a mtry of 5.

An advantage of using Random Forest is that it gives a list of the most important to the least important attributes in the dataset. Since Random Forest uses decision trees, the importance of each attribute needs to be calculated, so the algorithm can know which of the randomly selected attributes should be chosen for the split point. Thus after the final Random Forest algorithm is decided we can look at the list (see Table 19) and see that for our data, Violent Crime is the most important attribute and Rape is the least important, and there is a huge drop from Violent Crime to Robbery. This isn't that surprising as our data visualization revealed that Murder and Violent Crime were very the most correlated.

Finally, once the prediction outputs were created from the finalized Random Forest model, the outputs were compared to the actual murder data, and we could see the efficiency of the algorithm compared to Linear Regression. For evaluation, we calculated the Mean Square Error (MSE), Root Mean Square Error (RMSE), Man Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and a Confusion Matrix was constructed. For the confusion matrix we first turned the resulting data into intervals, before constructing the matrix.

Experiment/Result of 5-fold Cross-Validation on Linear Regression

Case 1: formula <- Murder~. (against all attributes)
(For more detailed information see Table 14.1.)

MAE: 0.000000000005339077927
MSE: 5.6214468979e-25
RMSE: 0.000000000000749763089 11
MAPE: 0.000014421944758 %
Accuracy: **99.999985578 %**
Confusion Matrix accuracy: **0.93335454** (Range: 0-3, 3-10, 10- 30, 30-80)
Confusion Matrix accuracy: **0.82312709** (Range: 0-1, 1-10, 10- 30, 30-80)

Case 2: formula <- Murder ~ Population+ Violent+ Robbery+ Burglary+Rape+ Assault+ PropertyCrime

MAE: 0.000000000000510611419 68
MSE: 5.4071501678e-25 RMSE:
0.000000000000735333269 19 MAPE:
0.000013200299335 % Accuracy:
99.9999868 % Confusion Matrix
accuracy:**0.93335454**(Range: 0-3, 3-10, 10- 30, 30-80)
Confusion Matrix accuracy:**0.80515349**(Range: 0-1, 1-10, 10- 30, 30-80)

Case 3: formula <- Murder ~ Violent+ Robbery+ Burglary+ Arson+Rape+ Assault+ PropertyCrime+ Theft+ SQmiles

MAE: 0.000000000000677035311 47 MSE:
6.3740991771e-25
RMSE: 0.000000000000798379557 42
MAPE: 0.000035696965786 %

```
Accuracy: 99.999964303 %
Confusion Matrix accuracy: 0.94862415 (Range: 0-3, 3-10, 10- 30, 30-80)
Confusion Matrix accuracy:0.8204231 (Range: 0-1, 1-10, 10- 30, 30-80)
```

Case 4: formula <- Murder ~ Violent+ Robbery+ Burglary+Arson+Rape+ Assault+ PropertyCrime+ Theft (For more detailed information see Table 14.2.)

```
MAE: 0.00000000000065546036106
MSE: 6.1599622623e-25
RMSE: 0.00000000000078485427069
MAPE: 0.000034438943274 %
Accuracy: 99.999965561 %
Confusion Matrix accuracy: 0.95085096 (Range: 0-3, 3-10, 10- 30, 30-80)
Confusion Matrix accuracy: 0.82312709 (Range: 0-1, 1-10, 10- 30, 30-80)
```

By running function "(modelLR)" in R, we obtained below information about our model, which gave detailed information on the algorithm performed:

```
Linear Regression:
6288 samples
  12 predictor
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 5031, 5031, 5030, 5031, 5029
Resampling results:

  RMSE                        Rsquared  MAE
  0.000000000000006059846112  1         0.000000000000040010840941
```

Experiment/Result of 5-fold Cross-Validation on Random Forest Algorithm

```
"MSE: 0.608816240108165"
"RMSE: 0.78026677496108"
"MAE: 0.430177415006739"
"R-squared: 0.964929268961206"
"MAPE: 19.1217789609063 %"
"Accuracy: 80.8782210390937 %"
"Confusion Matrix Accuracy: 89.13 %"
```

All datasets (raw, cleaned, normalized, etc), codes for cleaning, visualization, and algorithms can be found on GitHub: https://github.com/TML777/Crime_Rate_DM

Literature Survey/ Future Work: Related Works on Linear Regression

In the paper by Cahil and Mulligan, local crime patterns were explored by using Geographically Weighted Regression. It is another type of regression that helps to predict the undermining of the geographical location of the crime. The study shows that certain predictors of crime(racial distribution, economic status) can influence the outcome of the prediction(the research was done across Portland's(Oregon) various neighborhoods). Our data on the other hand did not include such features as race, indices of economic status, etc. By obtaining such information mentioned before, we can explore more on our topic of Crime(murder) prediction. One of the future work for our

project can be: drilling down the cities into neighborhoods, obtaining more information about people's race, nationality, income status, etc, and then trying to predict crime(murder). (M. Cahill, 2007)

The paper *"Crime Forecasting: a Machine Learning and Computer Vision Approach to Crime Prediction and Prevention"* by Neil Shah, etc, presents an analysis of using machine learning and computer vision techniques to predict and (potentially)prevent crimes. It underscores the application of various regression techniques in predicting crime patterns and occurrences: logistic regression, linear regression, KNN, and other ML algorithms. (Shah, 2021)

The paper "*Crime Prediction Model using Deep Neural Networks"* by Soon Ae Chun etc. investigates the feasibility of using deep neural networks to predict individual-level criminal behavior based on historical arrest data. The regression aspect in this context is embedded within the neural network's architecture, which utilizes historical data to forecast future outcomes, categorizing the potential severity of future offenses. As another idea for our future work, we could adapt Neural Network to our project. We will need more details (more features we have) and a huge number (1mln, not 6000) of observations. We also had to assign the weight for each feature, and then try to come up with a new NN algorithm that will help us to solve the problem efficiently. (S. Chun, 2019)

Literature Survey/ Future Work: Related Works on Random Forest (By Tigran)

An interesting paper by Varshika Gautem, Vidhi Yadav, and Sunil Kumr, titled *Diagnosis and Forecast of Murder rates in India using Random Forest and prophet algorithm*. They highlight the rise in crime in India, and the need to get a picture of the effects of different crimes. Once a better understanding is had, these can be deployed by law enforcement to "better manage police resources." Their aim was to identify motives for murder and kidnapping and they used a random forest classification algorithm, they also used a prophet algorithm, which "method for forecasting time series data using an additive model to meet nonlinear patterns."

Another paper worth mentioning is *Intelligent Crime Investigation Assistance Using Machine Learning Classifiers on Crime and Victim Information*. Here the objective was to use data collected from the Bangladesh Police, on specific crimes, to predict the attributes of the criminal. Specifically, the features used to predict were Area of Crime, Type of Crime, Victim Sex, Victim Race, Number of Victims, the targets were: Criminal Age, Criminal Sex, Criminal Race, and Methods of the Crimes. What's interesting about this paper is that they used 4 techniques to predict, what is essentially a classification problem. They used K-nearest Neighbor, Logistic Regression, Random Forest Classifier, and Decision Tree Classifier. Random Forest outperforms the other 3 in all cases except Criminal Sex, in which case it comes to a close second. Now Random Forest outperforming Decision Tree is not surprising, since Random Forests tend to handle overfitting better, but the other two show the potential random forest has in predicting crime.

Both papers end up going with a classification problem, whereas we are going with a regression problem. The second paper actually had to add labels to certain attributes, in order for the attribute to help accurately predict. The aim of both was to help law enforcement and to help lower crime in their respective fields. While the datasets for both were small, 1676 and 1466 respectfully, both papers show a promise in using machine learning, more specifically, Random Forests to help predict and fight crime.

Comparison and Conclusion

The project's result is successful since both algorithms gave us decent predictions with high accuracy. The Linear Regression outperformed the Random Forest by 18.99%. We showed in the experiment section the overall accuracy of the LR model was 99.99%, while RF had ~81%. There were no surprises in the results due to main two factors: first, it is not a common technique to use classification algorithms for predicting numerical values; second,

our data had a high correlation between attributes and class labels, which means our data set leaned in favor of Linearer Regression. According to many papers we explored, in highly correlated data, linear regression is the best algorithm to use (especially while predicting numerical values). The statistically significant attributes in predicting the “Murder” were “Violent”, “Assault”, and “Robbery”. These 3 attributes were the main predictors of the class label according to LR and RF algorithms.

Each step of the project starting from brainstorming about the topic, scraping the data, up to writing the paper taught us new skills. From this project, we understood that we can mine and find patterns for any meaningful data. We were able to learn how to clean, transform, and fill the gaps in the dataset. We learned how to create the model, train/test data, and evaluate our results. Of course, we faced many challenges, like learning the data preparation process in detail, exploring the new algorithms, dividing the work, reading related scientific literature(which was harder than expected), etc. Despite all, we were able to overcome challenges and learn many new skills previously unknown to us. In summary, the project was a success, and we were able to predict the number of murders in California cities with high precision.

Appendix

Table 1. Initial Downloaded Data for 2019

B	C	D	E	F	G	H	I	J	K	L	M	N
X	City	Population	Violentcrir	Murder.an	Rape1	Robbery	Aggravatec	Propertycr	Burglary	Larceny.th	Motorvehic	Arson
1	Adelanto	34,491	276	1	20	42	213	459	136	209	114	14
2	Agoura Hil	20,490	21	0	6	4	11	306	66	223	17	0
3	Alameda	78,907	162	0	7	94	61	2,579	218	1,958	403	29
4	Albany	20,083	40	0	8	21	11	685	105	534	46	1

Table 2. All samples(contains separate 15 data table within the list)

Name	Type	Value
allSamples	list [15]	List of length 15
crimeCalifornia2005	list [456 x 14] (S3: data.frame)	A data.frame with 456 rows and 14 columns
crimeCalifornia2006	list [457 x 14] (S3: data.frame)	A data.frame with 457 rows and 14 columns
crimeCalifornia2007	list [458 x 14] (S3: data.frame)	A data.frame with 458 rows and 14 columns
crimeCalifornia2008	list [459 x 14] (S3: data.frame)	A data.frame with 459 rows and 14 columns
crimeCalifornia2009	list [460 x 14] (S3: data.frame)	A data.frame with 460 rows and 14 columns

Table 3. CombinedAllSamples (30columns, ~6,900 tuples)

```
combinedAllSamples                          6896 obs. of 30
    $ City                                        : chr
    $ Population                                  : num
    $ Violent.crime                               : num
    $ Murder.and.nonnegligent.manslaughter        : num
    $ Forcible.rape                               : num
    $ Robbery                                     : num
    $ Aggravated.assault                          : num
    $ Property.crime                              : num
    $ Burglary                                    : num
    $ Larceny.theft                               : num
    $ Motor.vehicle.theft                         : num
    $ Arson1                                      : num
    $ Years                                       : num
    $ Violent..crime                              : num
    $ Murder.and..nonnegligent..manslaughter      : num
    $ Larceny..theft                              : num
```

Table 4. Summary of Data after Removing Population NA's

```
> mergedMurder <-combinedAllSamples
> summary(mergedMurder)
   Population         Robbery         Burglary          Years          Arson          NameCity
 Min.   :     89   Min.   :    0   Min.   :    0.0   Min.   :2005   Min.   :   0.00   Length:6896
 1st Qu.:  11982   1st Qu.:    5   1st Qu.:   63.0   1st Qu.:2008   1st Qu.:   1.00   Class :character
 Median :  31234   Median :   22   Median :  162.0   Median :2012   Median :   4.00   Mode  :character
 Mean   :  68054   Mean   :  114   Mean   :  380.3   Mean   :2012   Mean   :  15.67
 3rd Qu.:  70389   3rd Qu.:   75   3rd Qu.:  372.0   3rd Qu.:2016   3rd Qu.:  12.00
 Max.   :4029741   Max.   :14353   Max.   :22592.0   Max.   :2019   Max.   :2356.00
                                   NA's   :3                        NA's   :38
    Violent            Murder             Rape            Assault        PropertyCrime          Theft
 Min.   :    0.0   Min.   :  0.000   Min.   :   0.00   Min.   :    0.0   Min.   :     0.0   Min.   :    0
 1st Qu.:   29.0   1st Qu.:  0.000   1st Qu.:   2.00   1st Qu.:   18.0   1st Qu.:   273.0   1st Qu.:  168
 Median :   86.0   Median :  1.000   Median :   6.00   Median :   53.0   Median :   721.5   Median :  463
 Mean   :  310.9   Mean   :  3.396   Mean   :  18.88   Mean   :  174.5   Mean   :  1897.3   Mean   : 1189
 3rd Qu.:  227.0   3rd Qu.:  2.000   3rd Qu.:  16.00   3rd Qu.:  135.0   3rd Qu.:  1876.0   3rd Qu.: 1193
 Max.   :31767.0   Max.   :489.000   Max.   :2528.00   Max.   :17216.0   Max.   :117285.0   Max.   :67963
 NA's   :5                           NA's   :5         NA's   :5         NA's   :6
>
```

Table 5. Population Histogram (Log transformed)

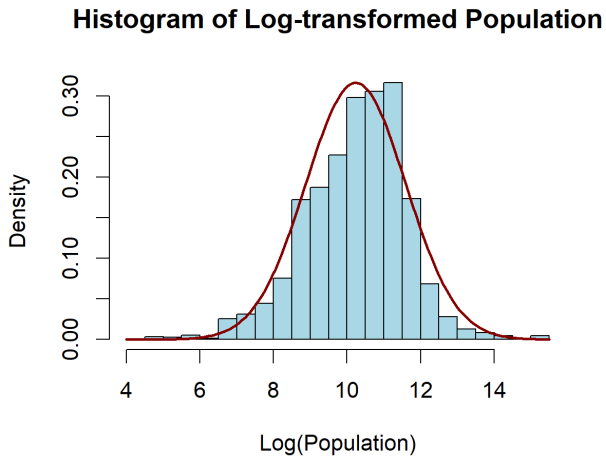


Table 6. Arson Histogram

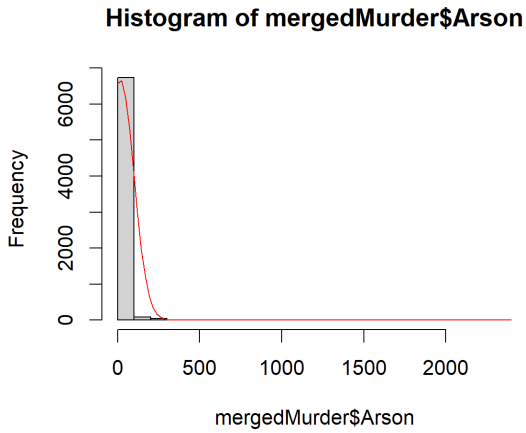


Table 7. Area Table

	City	SQmiles
1	City / Population	Land Area ▼
2	Los Angeles, CA / 3,862,210	468.67 sq mi
3	San Diego, CA / 1,341,510	325.19 sq mi
4	California City, CA / 13,243	203.52 sq mi
5	San Jose, CA / 986,320	176.53 sq mi
6	Bakersfield, CA / 358,700	142.16 sq mi
7	Fresno, CA / 506,132	111.96 sq mi
8	Palmdale, CA / 155,810	105.96 sq mi
9	Lucerne Valley, CA / 5,767	105.59 sq mi
10	Sacramento, CA / 476,075	97.92 sq mi
11	Lancaster, CA / 159,092	94.28 sq mi

Table 8. Square miles Features Before Cleaning

```
    SQmiles
 Min.   :  0.41
 1st Qu.:  3.76
 Median :  8.73
 Mean   : 17.45
 3rd Qu.: 19.47
 Max.   :468.67
 NA's   :139
```

Table 9. Boxplot of Population without/with outlier

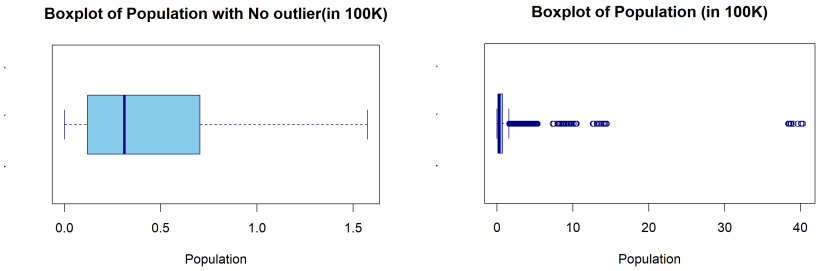


Table 10. Normalized Data Sample

	Population	Robbery	Burglary	Years	Arson	Violent	Murder	Rape	Assault	PropertyCrime
1	0.0073475327	0.00271720198	0.0158905807	0.35714286	0.0055178268	0.00761796833	0.008179959	0.0035601266	0.0110362454	0.0061900499
2	0.0083391816	0.00334424859	0.0084100567	0.85714286	0.0055178268	0.00749205150	0.004089980	0.0110759494	0.0092936803	0.0049281664
3	0.0071048319	0.00229917091	0.0156692635	0.14285714	0.0046689304	0.00481631882	0.010224949	0.0027689873	0.0062732342	0.0064628895
4	0.0072276713	0.00209015537	0.0154036827	0.21428571	0.0050933786	0.00582365348	0.004089980	0.0035601266	0.0083643123	0.0070085689
5	0.0085218277	0.00278687382	0.0077461048	0.92857143	0.0080645161	0.00752353071	0.000000000	0.0098892405	0.0101068773	0.0052010061
6	0.0079532922	0.00236884275	0.0181480170	0.42857143	0.0063667233	0.00516259011	0.002044990	0.0071202532	0.0064474907	0.0075457220
7	0.0085372136	0.00292621752	0.0060198300	1.00000000	0.0059422750	0.00868826140	0.002044990	0.0079113924	0.0123722119	0.0039135439
8	0.0082860753	0.00334424859	0.0106674929	0.78571429	0.0063667233	0.00868826140	0.002044990	0.0110759494	0.0115590149	0.0047746941
9	0.0080480895	0.00313523305	0.0175725921	0.50000000	0.0063667233	0.00626436239	0.002044990	0.0071202532	0.0078415428	0.0078782453
10	0.0060774479	0.00243851460	0.0105347025	0.07142857	0.0029711375	0.00428117229	0.004089980	0.0055379747	0.0049372677	0.0059001577
11	0.0074338926	0.00278687382	0.0153151558	0.28571429	0.0055178268	0.00856234457	0.004089980	0.0031645570	0.0128949814	0.0067954129

Table 11. Correlation matrix (sample of value representation)

```
> correlation_matrix
               Population     Robbery    Burglary       Years       Arson     Violent      Murder        Rape
Population     1.00000000  0.92602651  0.94005562  0.011619147  0.92063992  0.949269037  0.90465240 0.90192255
Robbery        0.92602651  1.00000000  0.93486385 -0.019175529  0.94860913  0.981028438  0.97372097 0.84594109
Burglary       0.94005562  0.93486385  1.00000000 -0.045532601  0.90571773  0.945179861  0.92452293 0.82473285
Years          0.01161915 -0.01917553 -0.04553260  1.000000000 -0.02115496 -0.009409549 -0.02359674 0.04936716
Arson          0.92063992  0.94860913  0.90571773 -0.021154964  1.00000000  0.946791123  0.94880084 0.82759767
Violent        0.94926904  0.98102844  0.94517986 -0.009409549  0.94679112  1.000000000  0.96297395 0.91619123
Murder         0.90465240  0.97372097  0.92452293 -0.023596742  0.94880084  0.962973946  1.00000000 0.81635665
Rape           0.90192255  0.84594109  0.82473285  0.049367157  0.82759767  0.916191232  0.81635665 1.00000000
Assault        0.94120185  0.94612983  0.93446053 -0.007468278  0.92525431  0.990634717  0.93578720 0.93436532
PropertyCrime  0.95682114  0.94748743  0.97540128 -0.017408731  0.90614142  0.959055415  0.91468239 0.86864919
Theft          0.94640038  0.92879210  0.94615210 -0.001981273  0.88454983  0.941016421  0.88199852 0.86859765
SQmiles        0.84861810  0.72245055  0.83067226  0.001684722  0.74198342  0.770321147  0.72172195 0.74592887
CityNumber     0.04753297  0.03908190  0.05160344 -0.003402884  0.01831644  0.046486561  0.03631539 0.04697323
Density        0.15304335  0.16615433  0.14206718  0.029771741  0.10266482  0.150273644  0.13780216 0.11888798
MurderRange    0.90348159  0.96830544  0.92432609 -0.023802370  0.94348148  0.958693818  0.99540804 0.81412781
```

Table 11. Correlation matrix (visual representation)

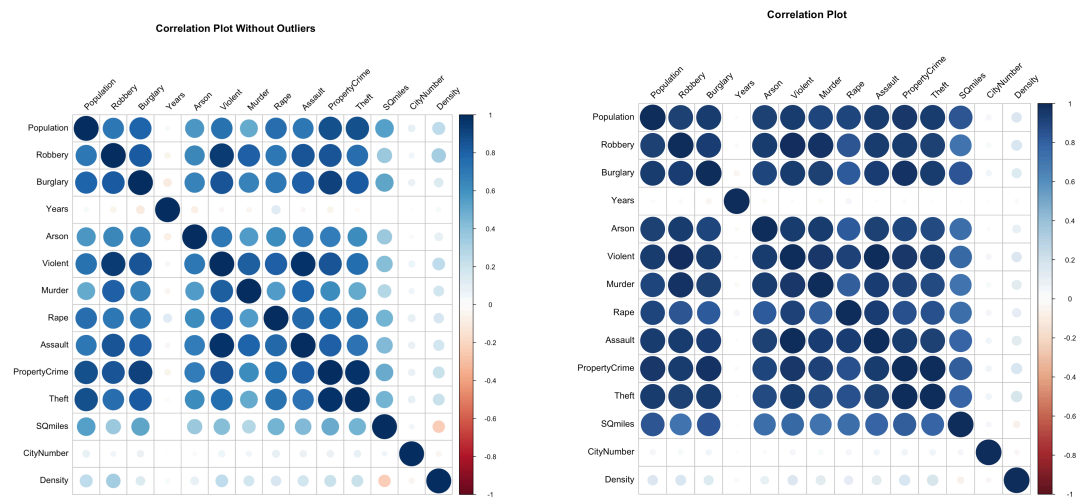


Table 12. Sample Scatter Plots and QQ-Plots Samples

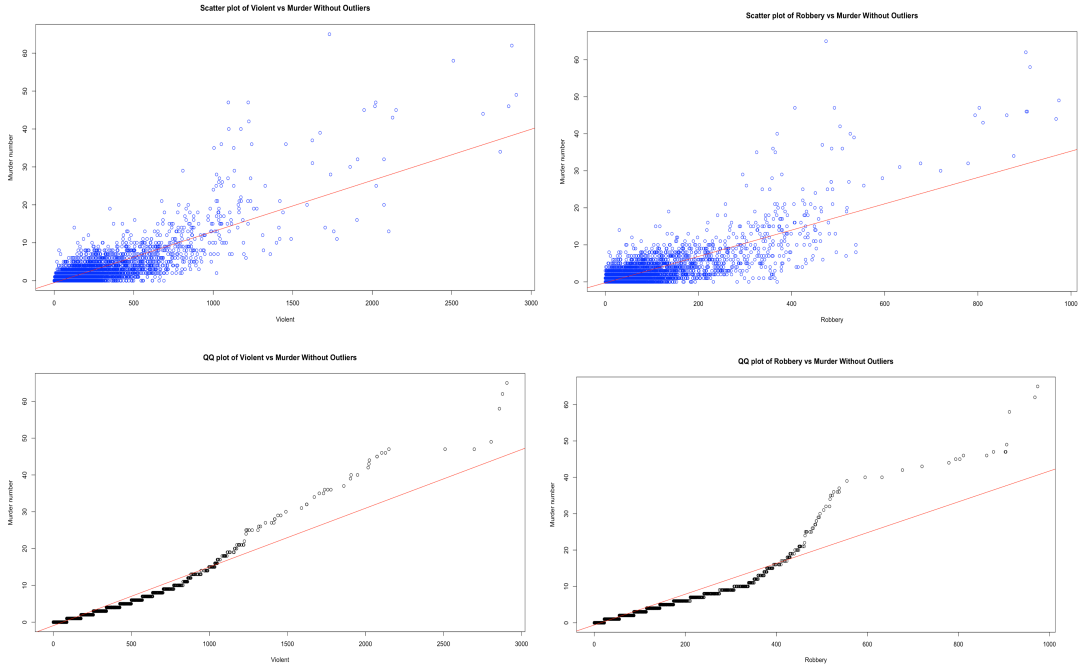
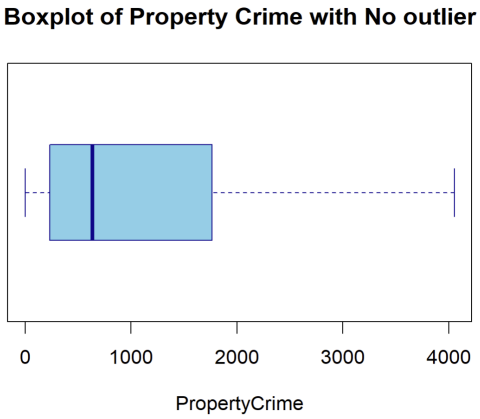
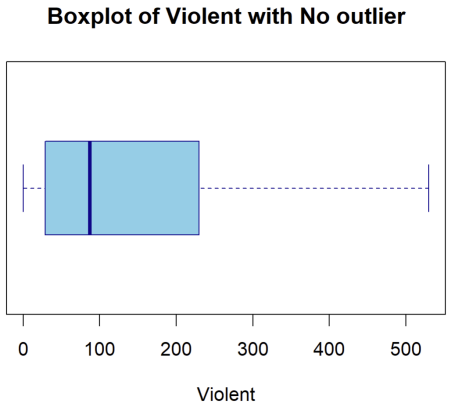
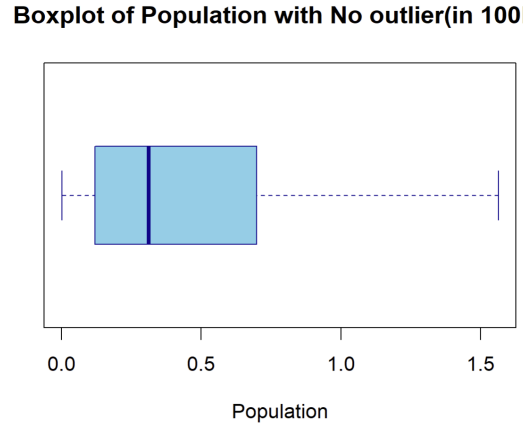


Table 13. Sample Boxplot







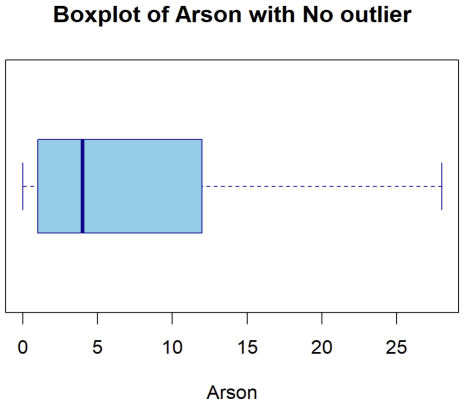


Table 14.1. Cross Validation 5-Fold Summary (case 1)

```
Confusion Matrix and Statistics

          Reference
Prediction 0 or 3  low medium high
    0 or 3   5031    0      0    0
    low       382  671      0    0
    medium      0   36    140    0
    high        0    0      1   26

Overall Statistics

               Accuracy : 0.93335454
                 95% CI : (0.92690644, 0.93939695)
    No Information Rate : 0.86098298
    P-Value [Acc > NIR] : < 2.22045e-16

                  Kappa : 0.77140318

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 or 3 Class: low Class: medium  Class: high
Sensitivity              0.92942915 0.94908062   0.992907801 1.0000000000
Specificity              1.00000000 0.93154122   0.994142532 0.9998402811
Pos Pred Value           1.00000000 0.63722697   0.795454545 0.9629629630
Neg Pred Value           0.69585987 0.99312190   0.999836361 1.0000000000
Prevalence               0.86098298 0.11245427   0.022427231 0.0041355177
Detection Rate           0.80022268 0.10672817   0.022268172 0.0041355177
Detection Prevalence     0.80022268 0.16748847   0.027994274 0.0042945761
Balanced Accuracy        0.96471458 0.94031092   0.993525167 0.9999201406
```

Table 14.2. Cross Validation 5-Fold Summary (case 4)

```
Confusion Matrix and Statistics

          Reference
Prediction 0 or 3  low medium high
    0 or 3   5119    0      0    0
    low       294  692      0    0
    medium      0   15    141    0
    high        0    0      0   26

Overall Statistics

               Accuracy : 0.95085096
                 95% CI : (0.94521485, 0.95606393)
    No Information Rate : 0.86098298
    P-Value [Acc > NIR] : < 2.22045e-16

                  Kappa : 0.82494327

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 or 3 Class: low Class: medium  Class: high
Sensitivity              0.94568631 0.97878359   1.000000000 1.0000000000
Specificity              1.00000000 0.94731183   0.997559388 1.0000000000
Pos Pred Value           1.00000000 0.70182556   0.903846154 1.0000000000
Neg Pred Value           0.74828767 0.99717035   1.000000000 1.0000000000
Prevalence               0.86098298 0.11245427   0.022427231 0.0041355177
Detection Rate           0.81421982 0.11006840   0.022427231 0.0041355177
Detection Prevalence     0.81421982 0.15683156   0.024813106 0.0041355177
Balanced Accuracy        0.97284316 0.96304771   0.998779694 1.0000000000
```

Table 15. Cross Validation: Visual representation

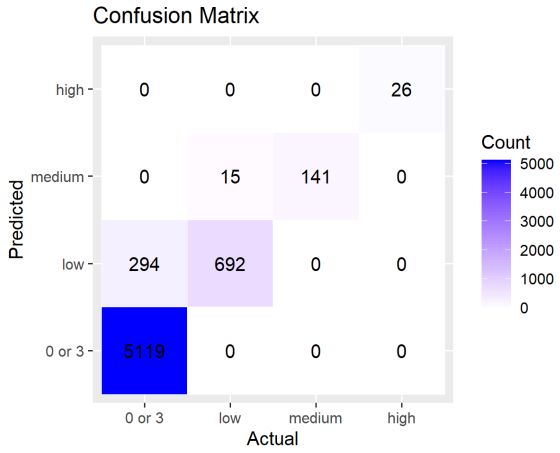


Table 16. Linear Regression

The dataset: $\{y_i,\ x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$

Where **y** is the target (label class), x_i are attributes, and β is the regression coefficient(a measure of the total effect of the predictor variables)

Model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i$$

i = 1, 2, …, n,

where ^{T} denotes the transpose, so that $x_i^T\beta$ is the inner product between vectors x_i and β, and ε_i is the i^{th} independent identically distributed normal error. (Linear Regression. Wikipedia)

Table 17. Number of Trees VS MSE

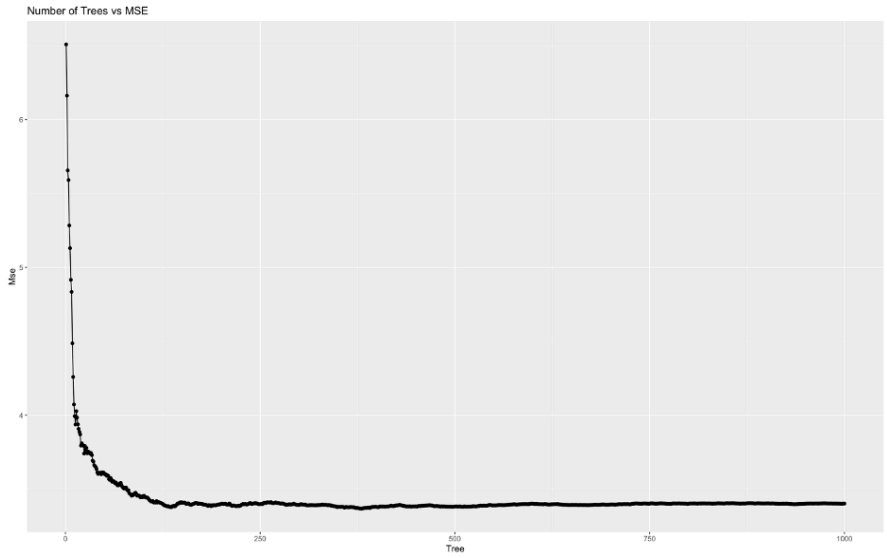


Table 18. Mtry Comparison 1:10

	mtry	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	1	1.915505	0.7758536	1.026560	0.2233059	0.04396384	0.05561283
2	2	1.860022	0.7852879	1.017678	0.1769859	0.03293316	0.05169909
3	3	1.851209	0.7869588	1.019853	0.1597497	0.02979185	0.05042968
4	4	1.847615	0.7878483	1.020679	0.1453453	0.02732317	0.04734661
5	5	1.847082	0.7882261	1.022831	0.1409833	0.02474072	0.04695352
6	6	1.851030	0.7876245	1.024002	0.1344879	0.02297482	0.04550555
7	7	1.852954	0.7874130	1.026063	0.1274865	0.02185773	0.04650042
8	8	1.859817	0.7859601	1.028236	0.1279782	0.02070695	0.04456948
9	9	1.858056	0.7867935	1.026075	0.1232412	0.01875030	0.04472329
10	10	1.854645	0.7870214	1.027237	0.1263919	0.01838821	0.04249963

Table 19. Random Forest Variable Importance

```
rf variable importance

              Overall
Violent       100.0000
Robbery        59.2716
Assault        30.0079
Burglary        7.8878
PropertyCrime   5.6624
Population      4.5514
Theft           3.5147
Arson           3.4622
SQmiles         2.9727
Years           0.1473
Rape            0.0000
```

References

FBI (n.d.). *Crime*. https://ucr.fbi.gov/crime-in-the-u.s

M. Cahill and G. Mulligan, "Using geographically weighted regression to explore local crime patterns", *Social Sci. Comput. Rev.*, vol. 25, no. 2, pp. 174-193, May 2007.

Manukyan, T., & Toleukhanova, A. (2024, April 22). *Crime_Rate_DM*. GitHub. Retrieved April 29, 2024, from https://github.com/TML777/Crime_Rate_DM

(n.d.). *Linear Regression*. Wikipedia. https://en.wikipedia.org/wiki/Linear_regression

(n.d.). *R Language Definition*. Cran.R-Project.org. https://cran.r-project.org/doc/manuals/r-release/R-lang.html

(n.d.). *Random Forest.* Wikipedia.org. https://en.wikipedia.org/wiki/Random_forest

(n.d.). *What Is Random Forest?*. IBM.com. https://www.ibm.com/topics/random-forest

S. Abdullah, F. I. Nibir, S. Salam, A. Dey, M. A. Alam and M. T. Reza, "Intelligent Crime Investigation Assistance Using Machine Learning Classifiers on Crime and Victim Information," *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, DHAKA, Bangladesh, 2020, pp. 1-4, https://ieeexplore-ieee-org.libproxy.csun.edu/document/9392668

S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim and G. R. Sinha, "An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach," in IEEE Access, vol. 9, pp. 67488-67500, 2021, doi: 10.1109/ACCESS.2021.3075140.

Shah, N., Bhagat, N. & Shah, M. Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention. *Vis. Comput. Ind. Biomed. Art* **4**, 9 (2021). https://doi.org/10.1186/s42492-021-00075-z

Soon Ae Chun, Venkata Avinash Paturu, Shengcheng Yuan, Rohit Pathak, Vijayalakshmi Atluri, and Nabil R. Adam. 2019. Crime Prediction Model using Deep Neural Networks. In Proceedings of the 20th Annual International Conference on Digital Government Research (dg.o 2019). Association for Computing Machinery, New York, NY, USA, 512–514. https://doi.org/10.1145/3325112.3328221

V. Gautam, V. Yadav and S. Kumar, "Diagnosis and Forecast of Murder rates in India using Random Forest and prophet algorithm," *2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, Ghaziabad, India, 2023, pp. 173-177, https://ieeexplore-ieee-org.libproxy.csun.edu/document/10141293