Adam Mills
Jacob Joiner
Neel Jhangiani
Patrick Wang

Math 4323 Final Report

**Introduction**
*written by: Adam Mills*

In this project, we explore the impacts of crime on university campuses across the United States. There are several types of crime committed that directly impact college students across America. Specifically, there are two major categories of crime: violent crime and property crime. Violent crimes consist of murder, rape, robbery, and aggravated assault, while property crimes consist of larceny, theft, arson, burglary, and motor vehicle theft.

> "In the FBI's Uniform Crime Reporting (UCR) Program, violent crime is composed of four offenses: murder and nonnegligent manslaughter, rape, robbery, and aggravated assault. Violent crimes are defined in the UCR Program as those offenses that involve force or threat of force. In the FBI's Uniform Crime Reporting (UCR) Program, property crime includes the offenses of burglary, larceny-theft, motor vehicle theft, and arson. The object of the theft-type offenses is the taking of money or property, but there is no force or threat of force against the victims. The property crime category includes arson because the offense involves the destruction of property; however, arson victims may be subjected to force. Because of limited participation and varying collection procedures by local law enforcement agencies, only limited data are available for arson. Arson statistics are included in trend, clearance, and arrest tables throughout Crime in the United States, but they are not included in any estimated volume data."[1]

Given a dataset describing seven different types of specific crime among American universities, we will use two supervised machine learning techniques, K-Nearest Neighbors (KNN) and Support Vector Machines (SVM), to explore the data and its impact. The project's main approach is to consider all of the crimes committed at universities by giving an overall assessment of the specific crime categories, and to predict whether a university has a higher property crime rate or a higher violent crime rate. This approach will aid a user in understanding both overall crime effects and the possibilities of crime being committed against universities in America. This project consists of several tasks; their step-by-step results are in the following section, and the corresponding code can be found in the appendix.

**Methodology**
*Written by: Adam Mills*

We explored two supervised learning techniques to classify crime in our dataset. The first approach leverages Support Vector Machines (SVM). This method is used for both regression and classification tasks; however, for this project, the focus is solely on classification. Support Vector Machines seek to separate labeled data from one another by enforcing a margin between samples from defined classes. The goal is to generate a decision boundary between classes with a large margin and minimal misclassifications. This method requires well-separated data. However, not all data is easily separated, and samples are easily misclassified. To address this

problem, we apply the kernel trick, which transforms data from one space to another to define better fit separation in a new space.[2] Different kernels transform data in different ways.

Given that beta and alpha are functions in the kernel,

$$\begin{cases} \hat{\beta}_0 = g_0(K(\mathbf{x}_1, \mathbf{x}_2), K(\mathbf{x}_1, \mathbf{x}_3)), \ldots, K(\mathbf{x}_{n-1}, \mathbf{x}_n)) \\ \hat{\alpha}_i = g_i(K(\mathbf{x}_1, \mathbf{x}_2), K(\mathbf{x}_1, \mathbf{x}_3), \ldots, K(\mathbf{x}_{n-1}, \mathbf{x}_n)),\ i \in S \\ \hat{\alpha}_i = 0,\ i \notin S \end{cases}$$

The general form of a hyperplane is:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

We apply kernels via the function K:

$$f(\mathbf{x}) = \beta_0 + \sum_{i \in S} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

There are several different kernels we address in this project.

Linear Kernel Equation:

$$\beta_0 + \sum_{k=1}^{p} (\sum_{i \in S} \alpha_i x_{ik}) x_k$$

Polynomial Kernel Equation:

$$\beta_0 + \sum_{i \in S} \alpha_i (1 + 2 \sum_{k=1}^{p} x_{ik} x_k + (\sum_{k=1}^{p} x_{ik} x_k)^2)$$

Radial Kernel Equation:

$$exp(-\gamma \textstyle\sum_{k=1}^{p} (x_{ik} - x_{jk})^2), \gamma > 0$$

Each hyperplane kernel equation transforms our data into a new space to search for the optimal separating hyperplane.

*Written by: Patrick Wang, Neel Jhangiani & Jacob Joiner*

Our second approach we used is KNN. KNN refers to K nearest neighbors. The KNN classifier identifies k points in the training data that are closest to other set points. We then classify the observation into a class with the highest estimated probability. The smaller the k value is, the more flexible the method will be. KNN produces highly accurate predictions, which do not need to be compared with other supervised learning methods. There is an absence of a training period, so we only use the training data set when we make the real-time predictions. This would then make KNN less demanding compared to other learning methods, as the two parameters necessary for KNN are the specified value of K and the distance function, such as Euclidean, Manhattan, etc.

SVM and KNN include a wide variety of differences. For example, SVM is less computationally demanding compared to KNN and is easier to interpret, but it can only identify a limited number of patterns. KNN does not assume a boundary, so it might be closer to the actual

relationship. For KNN, it has the advantage of classification fits that adapt to any boundary. SVM has the advantage of working with higher-dimensional spaces compared to KNN, as the time complexity is more memory efficient. However, both learning methods underperform when working with large datasets. Additionally, KNN is susceptible to outside noise, such as outliers and missing values, which may skew the overall prediction of the dataset that we use. SVM also has this issue, but it's when the target classes overlap.

**Data Analysis**

*Written by: Neel Jhangiani & Jacob Joiner*

In this project we pose our overall research question as, given a data set of university and crime statistics, can we predict the type of crime that occurs on each campus? Therefore, we will create a new variable labeled "crime", as our response variable. This variable will have two outputs; property crime and violent crime. We will label a university as "property crime" if the university's property crime is higher than the median property crime. We will label a university as "violent crime" if the university's property crime is lower than the median property crime. In the figure below we explore the two major types of crimes and their occurrences.

*Written by: Adam Mills*

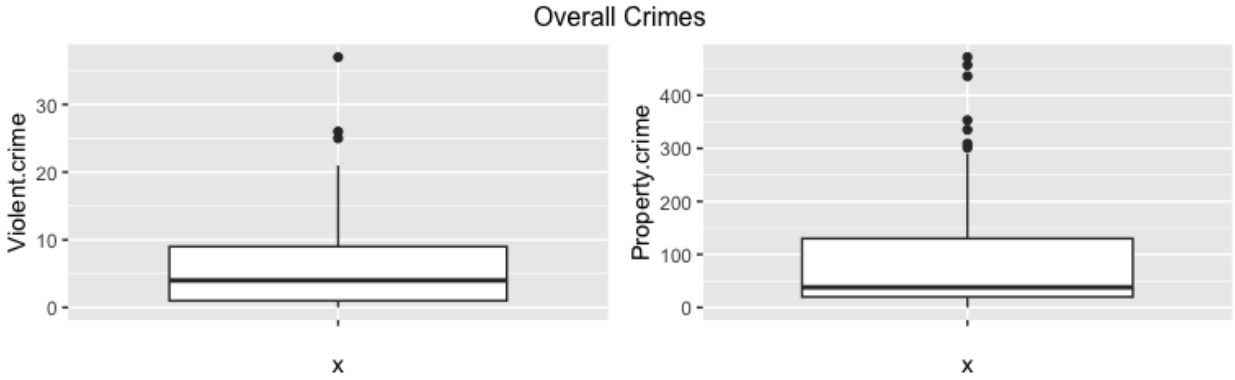


Figure 1: Boxplot of overall crime occurrences

In Figure 1 above, we visualize the occurrences of violent and property crimes across universities. Here we can see that property crimes occur far more often than violent crimes.

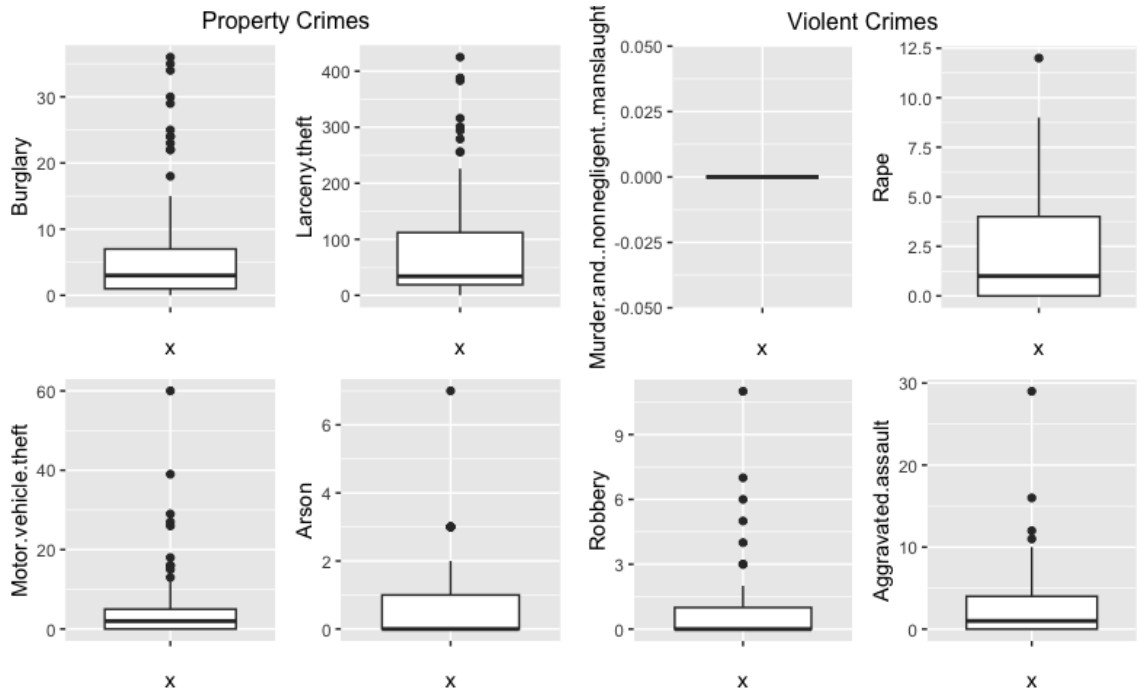


Figure 2: Boxplot of each specific crime occurrence

In Figure 2 above, we investigate each specific crime occurrence. On the left-hand side of Figure 2, we see the ranges of property crimes; the most widespread crime being larceny and theft, while cases of arson are not common. On the right-hand side, we explore violent crimes. Here, we note that there are no accounts of murder, while rape is the most common violent crime. For these reasons, we eliminate murder from our feature list as there are no occurrences.

Support Vector Machine *(written by Adam Mills)*

Support Vector Machines perform classification of a target variable by searching for the best decision boundary that maximizes the margin between all classes. In order to identify the best separating hyperplane, we perform the kernel trick, which transforms the data in different ways to look for separation in each space. The best-fitting model will provide the largest margin between classes. Here, we explore the effect of applying three different kernels to the data: linear, polynomial, and radial.

For the experiments, 10-fold cross-validation was used on our training set, which was created as a randomly sampled 80% of our data for the training set, and left 20% of the data as the testing set. Features used for analyzing crime are: student enrollment, rape, robbery, aggravated assault, burglary, larceny and theft, motor vehicle theft, and arson. The resulting features used for this dataset were chosen based on several experiments. For example, there are no accounts of murder, so this feature was eliminated as previously stated. Additionally, using each university's State as a feature was explored; however, this decreased the performance slightly. Furthermore, violent and property crime counts were eliminated, as the target variable is directly correlated to these two features.

Below in Table 1, several kernels were experimented with, such as Linear, Polynomial, and Radial Kernels. Also, scaling the data vs training with unscaled data was explored as well.

Table 1: Best Model Selection: 80:20 train-test split validation results

| | *Scaled Data* | *Unscaled Data* |
|---|---|---|
| *Training Error Linear Kernel* | 2.5% | 2.5% |
| *Test Error Linear Kernel* | 28.5% | 4.7% |
| *Training Error Polynomial Kernel* | 3.8% | 2.5% |
| *Test Error Polynomial Kernel* | 28.8% | 9.5% |
| *Training Error Radial Kernel* | 2.5% | 1.2% |
| *Test Error Radial Kernel* | 33.3% | 19.4% |

From Table 1 above, it is clear that the Linear Kernel without scaling the data provided the best model because the training and testing errors are lowest. In the case of the Radial Kernel, even though the training error is low, it is important not to use this model, as the test error is very high and performs poorly on unseen data.

The best fitting model using a linear kernel and a cost of 10. Using this model, the entire dataset is fit without splitting, and the results are shown in Table 2 below.

Table 2: Best Model Summary Using Full Dataset

```
Call:
svm(formula = y ~ ., data = scaled_data.train, kernel = "linear", cost = 10)

Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  10

Number of Support Vectors:  11

 ( 6 5 )

Number of Classes:  2

Levels:
 Property Crime Violent Crime
```

*Total Error:* 1.98%

The output of Table 3 shows us that there are 11 support vectors required to separate our two classes. The final error is 1.98%, meaning most of the samples are classified correctly. In Table 3 below, the confusion matrix is demonstrated on the full set.

Table 3: Confusion Matrix for the best Model on the Full Dataset:

| | *Property Crime* | *Violent Crime* |
|---|---|---|
| *Property Crime* | 48 | 2 |
| *Violent Crime* | 0 | 51 |

Table 3 above shows that violent crime is correctly classified every single time, while property crime had two instances of misclassification. To investigate why property crime is more difficult to predict, we explore the support vectors for each class in Table 4 below.

Table 4: Support Vectors for Each Class

| *SV #* | *Student Enrollment* | *Rape* | *Robbery* | *Aggravated Assault* | *Burglary* | *Larceny Theft* | *Motor Vehicle Theft* | *Arson* |
|---|---|---|---|---|---|---|---|---|
| *22* | Violent | Violent | Violent | Violent | Violent | Violent | Violent | Property |
| *100* | Violent | Violent | Violent | Violent | Violent | Violent | Violent | Violent |
| *47* | Violent | Violent | Property | Property | Violent | Property | Violent | Violent |
| *27* | Violent | Property | Violent | Violent | Property | Violent | Violent | Violent |
| *84* | Property | Property | Violent | Violent | Violent | Violent | Violent | Violent |
| *75* | Violent | Violent | Violent | Violent | Violent | Violent | Violent | Violent |
| *10* | Violent | Violent | Property | Property | Violent | Violent | Violent | Property |
| *96* | Property | Property | Violent | Violent | Violent | Violent | Violent | Violent |
| *50* | Violent | Violent | Violent | Violent | Violent | Violent | Violent | Property |
| *86* | Violent | Violent | Violent | Violent | Violent | Violent | Violent | Violent |
| *13* | Violent | Property | Violent | Violent | Violent | Violent | Violent | Violent |

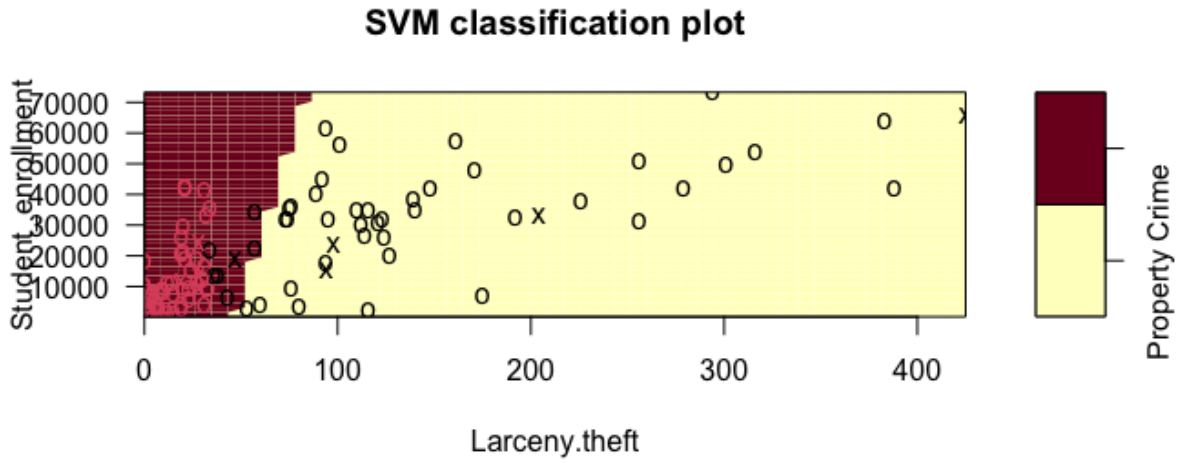


Figure 3: SVM Classification plot using two features

Both Table 4 and Figure 3 together demonstrate that there are more violent crime support vectors present compared with property crime support vectors. Therefore, when the data is learning to be classified, the support vectors tend to separate violent crimes better than property crimes. This is because the violent crimes are closely grouped, while the property crimes have more variation in the data.

K-Nearest Neighbors (*written by Neel Jhangiani & Jacob Joiner*)

For our KNN approach, we will use the last one out cross validation technique, with our newly created "crime" variable as our testing set. We decided to use the LOOCV method as it provides a less biased test misclassification error rate.

```
# Leave One Out Cross Validation#
set.seed(1)

knn.cv.pred <- knn.cv(train = uni_stats[,-12],
                      cl = uni_stats$crime,
                      k=1)
table(knn.cv.pred, uni_stats$crime)


knn.cv.pred      property crime violent crime
  property crime            133            46
  violent crime              57           145
> |
```

Using KNN with a K value of 1, we get a training error of 0.2703412.

Next, we will run the KNN function with six different K values.

```
for(k in c(1,3,5,10,15,20)) {
  set.seed(1)
  knn.cv.pred <- knn.cv(train = uni_stats[,-12],
                        cl = uni_stats$crime,
                        k=k)
  print(table(knn.cv.pred, uni_stats$crime))
  print(mean(knn.cv.pred != uni_stats$crime))
}
```

| K value | Training Error Rate |
|---|---|
| 1 | 0.2703412 |
| 3 | 0.2782152 |
| 5 | 0.2440945 |
| 10 | 0.2335958 |
| 15 | 0.2257218 |
| 20 | 0.2257218 |

From the table above, we can see that a k value of 10 gave us an optimal training error rate.

Confusion Matrix For Best KNN Model

| Knn.cv.pred | Property Crime | Violent Crime |
|---|---|---|
| Property Crime | 142 | 38 |
| Violent Crime | 48 | 153 |

Model Comparison: SVM vs KNN

Here we compare the best model results between KNN and SVM approaches. Support Vector Machine provided the best model error of 1.98%, leaving only two cases of class Property Crime misclassified as Violent Crime. K-Nearest Neighbors best model had an error of 22.57%, with 86 misclassifications out of 381 observations. Comparing the two models, SVM using the linear kernel gave us the best error rate. Both the KNN and SVM models predicted that universities have a higher property crime rate than violent crime rate.

**Conclusion** (*written by Patrick Wang*)

In this project, we analyzed the impacts of university crime in the United States using SVM and KNN supervised learning methods. Of the categories, the largest were violent and property crime. Using seven different kinds of crime from a dataset of American universities, we predicted whether a given university has higher crime rates in either of these two categories as well as the possibility of committing crime. Using our crime response variable, we labeled the universities as either violent or property based on which had higher rates at each university. First off, we have the SVM learning method, where we created a boxplot and found that overall crime occurrences lean more towards property crimes than violent crimes. We then generated a boxplot of each specific crime occurrence and found that larceny and theft were the main property crimes, while rape was the most common violent crime and no accounts of murder. Using the

three types of kernels such as linear, radial, and polynomial, we tested our data doing an 80:20 train-test split validation and found that the linear kernel method provided the best training and testing errors. Using this information, we then used the confusion matrix to test for the best model and found that property crime was harder to predict than violent crimes. To find the reason for this, we looked at the support vectors for each class and created an SVM classification plot and found that violent crimes were grouped closer together, while property crimes had a lot more variation. For the KNN learning method, we used LOOCV to input multiple different k-values and found the best training error rates for the dataset. We also included a confusion matrix for the KNN model.

Overall, the SVM least model error was 1.98%, with only two cases of property crime misclassified as violent crime, while KNN had the least model error of 22.57% with 86 misclassifications out of 381 observations. Out of both supervised learning methods, linear kernel SVM yielded the best error rate, yet both the models predicted that universities have higher property crime than violent crime rates. Some difficulties we faced while analyzing this data was choosing the right validation method for KNN. In the end, we collectively decided to use the Leave-one-out cross validation instead of k-fold cross validation to represent our data. With the linear kernel SVM, it was easy to use and had a lower complexity than the other kernel methods. The downside to this though was the need for proper training data to collect accurate results.

For future research on crime rates at universities in America, datasets could include more universities, as well as features that could increase or decrease crime rates. With these findings, we could then inform university administrations about their overall risk of violent or property crime rates and plan accordingly so they can provide more safety for students and staff. In conclusion, we found that property crimes were more prevalent than violent crimes at universities using SVM and KNN, with SVM resulting in the least error rate. These findings further show that data science is important for understanding and adapting to societal issues such as university crime in America.

**References**


1. FBI. “Property Crime.” *FBI*, FBI, 25 Aug. 2017, https://ucr.fbi.gov/crime-in-the-u.s/2016/crime-in-the-u.s.-2016/topic-pages/property-crime.
2. Sidharth. “SVM Kernels: Polynomial Kernel - from Scratch Using Python.” *PyCodeMates*, PyCodeMates, 12 Dec. 2022, https://www.pycodemates.com/2022/10/svm-kernels-polynomial-kernel.html.


**Appendix**

R-code

| |
|---|
| *Jacob Joiner & Neel Jhangiani Code:* Please see attached: Group_Project_KNN.R |
| *Adam Mills Code:* Please see attached: FinalProject_ALM.R and associated dataset |