Crime Prediction with Random Forest: Improving Law Enforcement with Data

Anju Sambasivan

Introduction

Random Forest is a powerful machine learning algorithm widely used for classification and regression tasks. It creates multiple decision trees, each trained on a random subset of data and features, and combines their predictions to improve accuracy. For classification, it selects the majority vote, and for regression, it averages the results. Known for its high accuracy, ability to handle missing data, and resistance to overfitting, Random Forest is a reliable tool for analyzing complex datasets.

In this project, a Random Forest model is applied to predict crime types using data such as the relationship between the victim and offender, location of the crime, age group, and police jurisdiction. By using these features, the model aims to assist law enforcement in identifying crime patterns, predicting crime types, and allocating resources more effectively. This project highlights the potential of machine learning to improve public safety and support smarter decision-making in law enforcement.

Data Features and Model Building

The dataset used for this project has 49,332 records. Missing values were removed using the `na.omit()` function in R. Some variables, like AgeGroup, LocationType, ROVDivision, PoliceArea, and AnzsocDivision, were converted into categorical variables (factors). The data was then split into 70% for training the model and 30% for testing.

Built the Random Forest model using 30 trees (`ntree = 10`). At each split, the model used 3 randomly selected features (`mtry = 3`). The target variable was AnzsocDivision, which represents the type of crime and also enabled an option to measure how important each predictor variable is for the model.

Assumptions

The dataset is assumed to reflect real-life situations, allowing the model to work effectively with new data. The model uses details such as the victim's age group, the location of the crime, whether the victim knew the offender, if it involved a person or an organization, and the police area. These factors provide insights into who was involved and where the crime occurred.

Missing values that were not significant for the results were removed. The "Territorial Authority" feature was excluded due to its high number of categories (68), which could add unnecessary complexity. Other geographic details in the data were sufficient for accurate predictions.

```
# Load necessary libraries
library(dplyr)
library(randomForest)
library(caret)
library(knitr)
```

```
# Read the dataset
data <- read.csv("data2.csv")
```

```
# Convert relevant columns to factors (categorical variables)
data$AgeGroup <- as.factor(data$AgeGroup)
data$LocationType <- as.factor(data$LocationType)
data$ROVDivision <- as.factor(data$ROVDivision)
data$TerritorialAuthority <- as.factor(data$TerritorialAuthority)
data$PersonOrOrganization <- as.factor(data$PersonOrOrganization)
data$AnzsocDivision <- as.factor(data$AnzsocDivision)
data$PoliceArea <- as.factor(data$PoliceArea)
```

```
# Print the structure of data
str(data)
```

```
# Convert the Date column to proper date format
data$Date <- as.Date(data$Date)
```

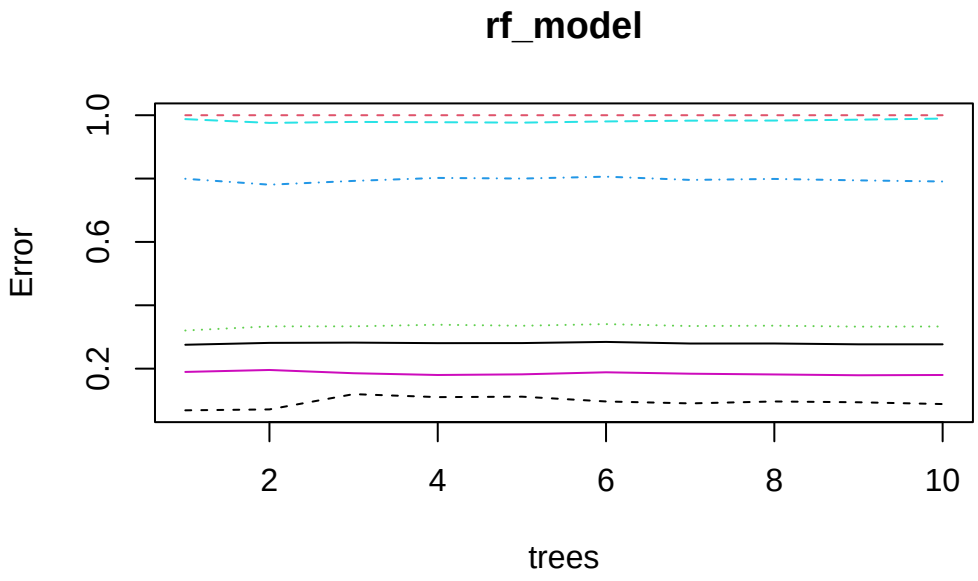
```
# Remove rows with missing values
data_clean <- na.omit(data)
```

```r
# Split the data into training (70%) and testing (30%) sets
set.seed(123)
trainIndex <- createDataPartition(data_clean$AnzsocDivision,
                                  p = 0.7, list = FALSE)
train_data <- data_clean[trainIndex, ]
test_data <- data_clean[-trainIndex, ]
```

```r
# Train a Random Forest model
rf_model <- randomForest(AnzsocDivision ~ AgeGroup +
                           LocationType + ROVDivision +
                           PersonOrOrganization + PoliceArea,
                           data = train_data,
                           ntree = 10,
                           mtry = 3,
                           importance = TRUE)
```

Error Plot

```r
# Plot the Random Forest model error rate as trees are added
plot(rf_model)
```



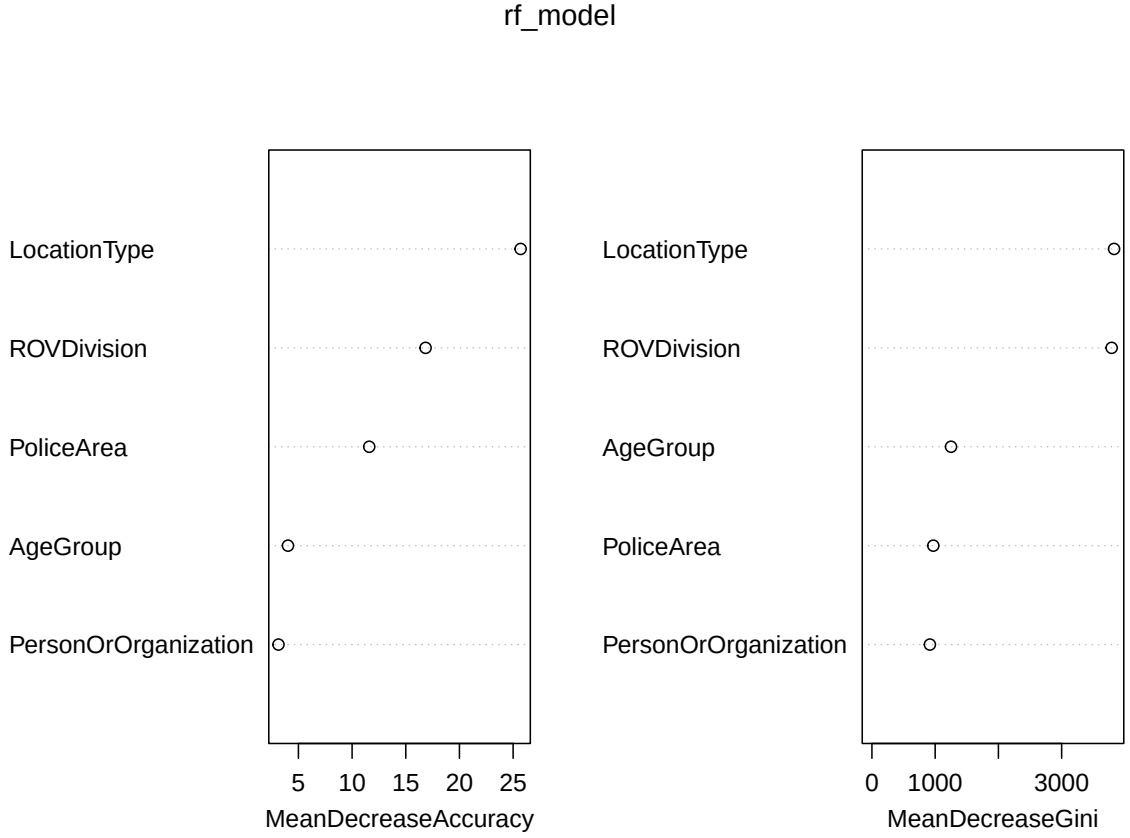
This graph shows how the Random Forest model's error rate changes as the number of trees increases from 1 to 10. The x-axis represents the number of trees in the model, while the y-axis shows the error rate, which measures how often the model makes incorrect predictions.

The black line represents the overall error across all crime categories. It stabilizes as more trees are added, meaning that after about 6 trees, the model's performance does not improve much. The colored lines show the error rates for specific crime categories. Some categories have consistently low error rates, indicating the model predicts them accurately. However, categories with higher error rates, closer to 1.0, suggest that the model struggles to predict those types of crimes correctly.

Overall, the graph shows that the model performs consistently after a certain number of trees, and adding more trees beyond this point does not significantly improve accuracy. The higher error rates for some categories highlight the need for additional data or better features to improve predictions for those crimes.

Variable Importance Plot

```
# Plot variable importance (how significant each variable is for prediction)
varImpPlot(rf_model)
```



The left graph shows how much the model's accuracy decreases when a specific feature is removed. This helps identify which features are the most important for making accurate predictions. The feature `LocationType` stands out as the most important, meaning that knowing the type of location, such as a public place or residential area, significantly helps the model predict crime types correctly. Other features, including `ROVDivision` (whether the victim knew the offender), `PoliceArea`, `AgeGroup`, and `PersonOrOrganization`, also contribute to the model's accuracy but are less important compared to `LocationType.`

The Gini Index is a method used in decision trees to measure how well a feature separates the data into clear groups. A clear split means that most of the data points in a group belong to the same category (like the same crime type). The Mean Decrease Gini measures how much a feature helps improve these splits across all trees in the Random Forest model. Features with higher Mean Decrease Gini values are more important because they help create better splits, which makes the model's predictions more accurate.

In the right graph, `LocationType` has the highest Mean Decrease Gini value, meaning it is the most useful feature for splitting the data into clear groups. For example, knowing whether a crime happened in a public place or residential area helps the model better predict the type

of crime. Other features, such as `ROVDivision` (relationship to the victim), `AgeGroup`, and `PoliceArea`, also help improve the splits, but they are not as important as `LocationType`. This panel shows that `LocationType` is the key feature for making accurate predictions, while the other features still add some value.

```
# Predict on the test dataset
rf_predictions <- predict(rf_model, newdata = test_data)
```

```
# Create a confusion matrix to evaluate the model's performance
conf_matrix_rf <- confusionMatrix(rf_predictions, test_data$AnzsocDivision)
```

```
# Convert the confusion matrix into a data frame for visualization
conf_matrix_df <- as.data.frame(conf_matrix_rf$table)
```

```
# Simplify Prediction labels
conf_matrix_df$Prediction <- sapply(strsplit(as.character(
  conf_matrix_df$Prediction), " "), `[`, 1)
```

Confusion Matrix

```
# Visualize the confusion matrix as a heatmap
ggplot(conf_matrix_df, aes(Prediction, Reference, fill = Freq)) +
  geom_tile(colour = "white") +
  scale_fill_gradient(low = "lightblue", high = "blue") +
  labs(x = "Predicted Class", y = "Actual Class", title = "Confusion Matrix")+
  theme_minimal() +
  geom_text(aes(label = Freq), size = 3) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 8),
        axis.text.y = element_text(size = 8),
        plot.title = element_text(size = 10, hjust = 0.5)
        )
```

The confusion matrix shows how well the Random Forest model predicted different crime types. The diagonal cells represent the correct predictions made by the model, while the off-diagonal cells show where the model made mistakes.

The model correctly predicted 5,713 cases of Theft and Related Offences, which is the highest number of correct predictions. It also correctly predicted 3,386 cases of Acts Intended to Cause Injury and 1,517 cases of Unlawful Entry with Intent/Burglary.

However, there were some misclassifications. For example, 1,089 cases of Theft and Related Offences were incorrectly predicted as Acts Intended to Cause Injury, and 1,679 cases of Acts Intended to Cause Injury were incorrectly predicted as Theft and Related Offences. This could be because these two types of crimes share similar patterns, such as the people involved, locations, or circumstances, making them harder for the model to distinguish. These errors suggest that the model could be improved with better tuning and additional features.

```r
# Extract overall model statistics
overall_stats <- data.frame(
  Statistic = c("Accuracy",
                "No Information Rate",
                "Kappa",
                "95% CI",
                "P-Value"),
  Value = c(
    round(as.numeric(conf_matrix_rf$overall['Accuracy']), 3),
    round(as.numeric(conf_matrix_rf$overall['AccuracyNull']), 3),
    round(as.numeric(conf_matrix_rf$overall['Kappa']), 3),
    sprintf("(%.3f, %.3f)",
            as.numeric(conf_matrix_rf$overall['AccuracyLower']),
            as.numeric(conf_matrix_rf$overall['AccuracyUpper'])),
    format.pval(as.numeric(conf_matrix_rf$overall['AccuracyPValue']))
  )
)
```

Overall Statistics

```r
# Display the statistics as a table
kable(overall_stats, caption = "Overall Statistics for the Random Forest Model")
```

Table 1: Overall Statistics for the Random Forest Model

Statistic	Value
Accuracy	0.725
No Information Rate	0.47
Kappa	0.555
95% CI	(0.717, 0.732)
P-Value	< 2.22e-16

1. **Accuracy (0.725):** Accuracy measures how often the model makes correct predictions overall. An accuracy of 0.725 (or 72.5%) means the model correctly predicted the crime type in about 73 out of 100 cases. It is a simple and widely used indicator of the model's performance.
2. **No Information Rate (0.47):** The No Information Rate (NIR) represents the accuracy that would be achieved by always guessing the most frequent class in the dataset. For example, if 47% of crimes in the dataset belong to one type, then predicting that type

for all instances would yield 47% accuracy. The model's accuracy (72.5%) being much higher than the NIR shows that it provides meaningful predictions beyond guessing the most common class.

3. **Kappa (0.555):** Kappa measures the agreement between the model's predictions and the actual results, considering the possibility of correct predictions occurring by chance. A Kappa value ranges from -1 to +1:

- **+1:** Perfect agreement between predictions and actual outcomes.
- **0:** Performance is no better than random guessing.
- **Less than 0:** Worse than random guessing.

A Kappa value of 0.555 indicates moderate agreement, meaning the model is better than random guessing but not perfectly accurate.

4. **95% Confidence Interval (0.717 to 0.732):** This interval provides a range in which the true accuracy of the model is expected to lie 95% of the time. For this model, it suggests that the accuracy is consistently between 71.7% and 73.2%, indicating stability and reliability in the model's performance.

5. **P-Value (< 2.22e-16):** The p-value tests whether the model's accuracy is significantly better than random guessing. A very small p-value (close to zero) indicates strong evidence that the model's performance is not due to chance. For this model, the extremely small p-value confirms that its good performance is statistically significant and reliable.

Conclusion

The Random Forest model is a useful tool for predicting different types of crimes using key details like the relationship between the offender and victim, the location of the crime, age group, and police station area. By using this model, police can better allocate their resources and focus on areas where they are needed the most.

The model also helps identify patterns in crimes, such as locations where certain crimes are more likely to happen. This can support preventive measures like increasing patrols, deploying more officers, or educating the community. Additionally, the model can assist in investigations by predicting possible crime types when the details are unclear, giving police a starting point.

Although the model performs well, there is room for improvement, especially in handling similar crime types. Adding more data and refining the model could make it even more accurate. Overall, this project shows how machine learning can support law enforcement and improve public safety through smarter, data-driven decisions.