PREDICTING CRIME IN THE NETHERLANDS: COMPARING MACHINE LEARNING ALGORITHMS

WIEBRAND VAN KESSEL

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2011951

COMMITTEE

dr. Drew Hendrickson
MSc. Büşra Özgöde Yigin

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

June 19th, 2023

WORD COUNT

8377

PREDICTING CRIME IN THE NETHERLANDS: COMPARING MACHINE LEARNING ALGORITHMS

WIEBRAND VAN KESSEL

Abstract

In the field of criminology, crime statistics have always been a matter to research and to do predictions on. Many studies have been conducted in this field, exploring the relations between other contexts' demographic, socio-economic, and environmental features. There is, however, still a need for research into the predictive capabilities of machine learning algorithms, especially when applied to data from the Netherlands. The aim of this study is to apply these algorithms to a dataset containing historic crime combined with regional demographic, socio-economic, and weather features. After applying these, the algorithms were tuned, and their results were evaluated. The algorithms used are Random Forest, XGBoost, K-nearest neighbors (KNN), and Multilayer Perceptron. These are compared to a baseline set by a linear regression. The results demonstrate that tree-based algorithms (XGBoost especially, with an MAE of 16.318 and an R-squared of 0.995) are the most precise in predictions on the dataset provided, with low errors and high R-squared. Multilayer Perceptron and K-nearest neighbors have lower performance, having higher errors and lower R-squared in predictions. The feature importance and error patterns of the XGBoost model were also analyzed and reviewed. Important findings in these analyses were the importance of the percentage of people with immigrant backgrounds and the percentage of single-parent families. These findings contribute to the literature on crime prediction in the Netherlands and help to gain insights for law enforcement and policymakers to assist in making communities safer.

1 INTRODUCTION

This study aims to find the best-performing machine learning algorithm for predicting crime in The Netherlands. Comparing them based on

socioeconomic and demographic factors, with the addition of weather data.

1.1 *Problem statement*

Investigation, intervention, and prevention are some of the main activities of the Dutch police, with prevention being the most desired outcome at all times. Growing crime rates force them and society to develop new ways to direct resources to the most beneficial areas. An accurate crime prediction allows police to respond to incidents more quickly and intervene more efficiently. This would also increase feelings of safety within the community, which is affected gravely by crime rates (Johansson & Haandrikman, 2021; Visser et al., 2013). The effectiveness of resource allocation is the main reason why it is essential to have accurate predictions. The Dutch police have limited resources, including funding, personnel, and equipment. This limitation creates a necessity for prioritization of the available resources based on a prediction of crime per area, to be more precise. This prediction per region, when accurate, could have a significant positive impact on this resource allocation, providing law enforcement with the funding and personnel where needed most.

The important aspects of accurate crime prediction in resource allocation are shown in multiple recent studies. One of these studies was done by Meijer and Wessels (2019). This study concluded that implementing such a predictive model positively affected most of their reviewed cases. Alves et al. (2018) also found the importance of this prediction to be substantial when investigating homicide predictions using urban metrics. A study on the effect of using predictive software in New York City (Levine et al., 2017) found that the software implications lead to significant financial savings and increased officers' productivity. However, crime predictions can also have drawbacks (Meijer & Wessels, 2019), such as the possibility of racial profiling (Goel et al., 2016). These drawbacks lead to a counter-intuitive outcome when predictions are made. Despite concerns, crime prediction through analysis of historical data and relevant features can help the Dutch police to allocate resources more efficiently. This would not only be in favor of the police themselves but also of the community, which can be patrolled and protected more effectively. Less unnecessary patrols through neighborhoods have also been found to reduce feelings of safety (Van de Veer et al., 2012). Research on crime prediction is not sparse; however, there are shortcomings in the existing literature. For example, studies on crime prediction have been conducted in cities and states in the US and India. Still, there is a need for more research to explore the implementation of such prediction models in The Netherlands and the

accuracy such models would provide. Also, the use of weather data has not been studied in The Netherlands, while it has shown potential in other research (Algahtany et al., 2022; X. Chen et al., 2015). This study aims to fill this gap in the existing literature by exploring the possibility of predicting regional crime statistics in The Netherlands. This will be done by using variables like historical and annually updated demographic figures on the socio-economic and demographic status of each of the municipalities in The Netherlands in combination with weather data, attempting to create a model that can accurately predict regional crime statistics in The Netherlands. Additionally, the importance of the features used in the model and the errors made by the model are evaluated to support future research into the subject.

1.2 *Research Questions*

The main research question for is study is defined as:

> *To what extent is it possible to predict future crime statistics in the Netherlands when using machine learning algorithms?*

The sub-questions can be listed separately as such:

RQ1 *How well do Random Forest, XGBoost, KNN, and Multilayer Perceptron perform when predicting crime?*

Since crime prediction has become increasingly important to law enforcement, it is important to determine which type of algorithm works best on the data and predicts the most accurately. Therefore, an evaluation is done on the performance of different machine learning algorithms, which is done by comparing them to the baseline score from a Linear Regression. The evaluated algorithms are Random Forest, XGBoost, KNN, and Multilayer Perceptron.

RQ2 *Which features are most important for the best-performing models when predicting crime?*

We now know what model performs best, but we also want to know what features the model relied on most. The benefit of identifying these is, first of all, that these can be used to build upon in this and future research, focusing more on the important features. Secondly, establishing leading factors for high crime rates is beneficial for law enforcement to distribute resources.

RQ3 *How does the best-performing model differ in performance when population size increases?*

This research question provides insight into the performance and effectiveness of the predicting model. Evaluating the performance over different population size levels provides information on how accurate the best-performing model is and how its errors are distributed. This information can be used to identify potential weaknesses of the model when applied to larger groups of people. Knowing what errors a model produces in crime prediction allows for improvements to the model and in future research.

2 RELATED WORK

The most important choices for crime prediction models are, first of all, the choice of algorithms to use and, second of all, the features to include for doing prediction. The field of crime prediction has been explored by many researchers, with many different takes on combining algorithms and features. This literature review is divided into two parts, where different studies in the field of crime prediction are discussed and compared to each other. Firstly, *algorithms for crime prediction* discusses the different algorithms used by studies in machine learning. Secondly, *Relevant features*, review the different features and types of features which have proven to be useful or have the potential to predict future crime statistics over different regions.

2.1 *Algorithms for crime prediction*

Across studies, various machine learning algorithms have been used, each with its benefits and drawbacks. To research which algorithms to include in this study, the most commonly used are reviewed for their performance in other studies. The algorithms which are taken into account are Multilayer Perceptron (MLP), Decision Tree (DT), Random Forest (RF), XGBoost, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Past research on crime prediction from regions across the world is taken into account when reviewed.

Shah et al. (2021) presented a study comparing KNN, Random Forest, and SVM for predicting crime in India, concluding that KNN, Random Forest, and SVM were all viable and effective options for crime prediction in that region. In a study by Tamilarasi and Rani (2020), crime against women in India was predicted using KNN and Decision Tree, after which the results were compared. The results were that KNN's performance was better than the Decision tree when assessing both the explainability and errors of each. Results from A. Kumar et al. (2020) came to the same conclusion when predicting time series data in the same region as

Tamilarasi and Rani (2020). The final study in India, which is considered, is the one conducted by Aziz et al. (2022). The aim of this study was to predict the total number of crimes committed across India. Random Forest, Decision Tree, and Multilayer Perceptron were mostly compared regarding R-squared. The R-squared of the Random Forest was 0.96, meaning that the model explained 96% of variability in the data. MLP was the second best in terms of fit to the data, scoring an R-squared of 0.89, while Decision Tree only had an R-squared of 0.57.

A study by Cavadas et al. (2015) using data from the United States found that Random Forest Regression had the best performance when predicting the number of crimes per hundred-thousand people. Random Forest outperformed SVM and Multivariate Adaptive Regression Splines (MARS) by a solid margin after using K-fold cross-validation. Even Though SVM did not present top-of-the-line results in the studies mentioned so far, Shamsuddin et al. (2017) did find SVM to be a very effective algorithm for predicting crime in Malaysia. Safat et al. (2021) also studied crime in the United States, reviewing the cities of Chicago and Los Angeles. Attempting to predict crime statistics in both cities accurately, they eventually determined that KNN and XGBoost were both equally well-performing options for their datasets. Decision Tree and MLP performed worst by some margin, which aligns with conclusions from other studies mentioned. The benefits of KNN, like easy treatment of missing values, are also mentioned by Malathi and Baboo (2011). To conclude the comparison of these algorithms, a study by Zhang et al. (2022) focused on two of the best-performing algorithms mentioned above, Random Forest and XGBoost. When predicting crime in Southeast China using these two, the results were that XGBoost was a far superior predictive algorithm in the study. Random Forest did outperform Logistic Regression and Decision Tree, but this makes sense when the other studies are reviewed.

In terms of results, machine learning algorithms have been used in many different studies and have produced promising outcomes. Despite the fact that studies were being conducted in different parts of the world, MLP, KNN, XGBoost, and Random Forest are very popular machine-learning algorithms with good results in crime prediction. SVM and Decision Trees are also used in many studies, but compared to the other algorithms, they could not compete regarding R-squared score and error.

2.2 *Relevant features*

Selecting models to do predictions with is important, but the features put into the model are the basis on which a study is built. To select which features to use for predicting crime in The Netherlands, past research is

reviewed to find features that have been important in similar studies or in other fields.

A study conducted by Jenga et al. (2023) researched many different machine learning papers, of which the majority used supervised machine learning (Decision Tree, SVM, KNN) and came to some overall conclusions. Firstly, Jenga et al. (2023) found that one of the biggest challenges in predicting crime is that, depending on which region you are in, not all features are measured, and not all crimes are reported. This leaves a hole in the data, making it impossible always to develop perfect models, no matter which model is selected.

In a study by Short et al. (2008), high levels of crime are linked to the percentage of the male population, which is supported in multiple other types of research like Archer (2022) and Cortoni et al., 2017. Safat et al. (2021), found that income, unemployment levels, and other socio-economic factors are useful predictors of crime. These were then used throughout the research, which compared, among others, KNN, XGBoost, Random Forest, and LSTM. The study by Alves et al. (2018) implemented Random Forest purely on homicides in Brazil, resulting in a high feature importance for the variables unemployment, illiteracy, and male population. The male population is the second most important, and unemployment is the most important predictive feature. The importance of including unemployment as a feature was stressed in a study by Kassem et al. (2019), which found there to be a direct link between rising unemployment and crime rates. This link was also drawn between crime and the variable of population density. A socio-demographic variable that has found support on both sides of the spectrum is the percentage of the population with an immigration background. For example, MacDonald et al. (2013) found that a higher percentage of the population with an immigration background significantly reduces the average amount of violent crimes. This same conclusion was also drawn by Xie and Baumer (2019) and Kubrin and Ishizawa (2012) when researching the entire United States and Los Angeles. Another interesting study by Zhang et al. (2022) used features such as proximity to police stations, population size, and richness to predict crime in Southeast China using XGBoost, resulting in a model that achieved an accuracy of 90%.

X. Chen et al. (2015) researched the relation between crime, Twitter sentiment, and weather in Charlottesville, USA. While no direct link with Twitter sentiment was established, the weather variables were significant in predicting crime. For example, a higher temperature increased the chances of theft, and low humidity lowered those same chances of theft. Algahtany et al. (2022) also studied the effects of humidity and temperature on crime rates and concluded that there was a significant relationship. However,

this study was conducted in the Middle-east, meaning this relation could be non-existent in a different region such as The Netherlands. In Boston, which has a more comparable climate to The Netherlands, Sommer et al. (2018) observed a significant increase in crimes when the temperature rose from cold to moderate. Sommer et al. (2018) also observed that rainfall decreased crime rates in Boston.

3 METHODOLOGY

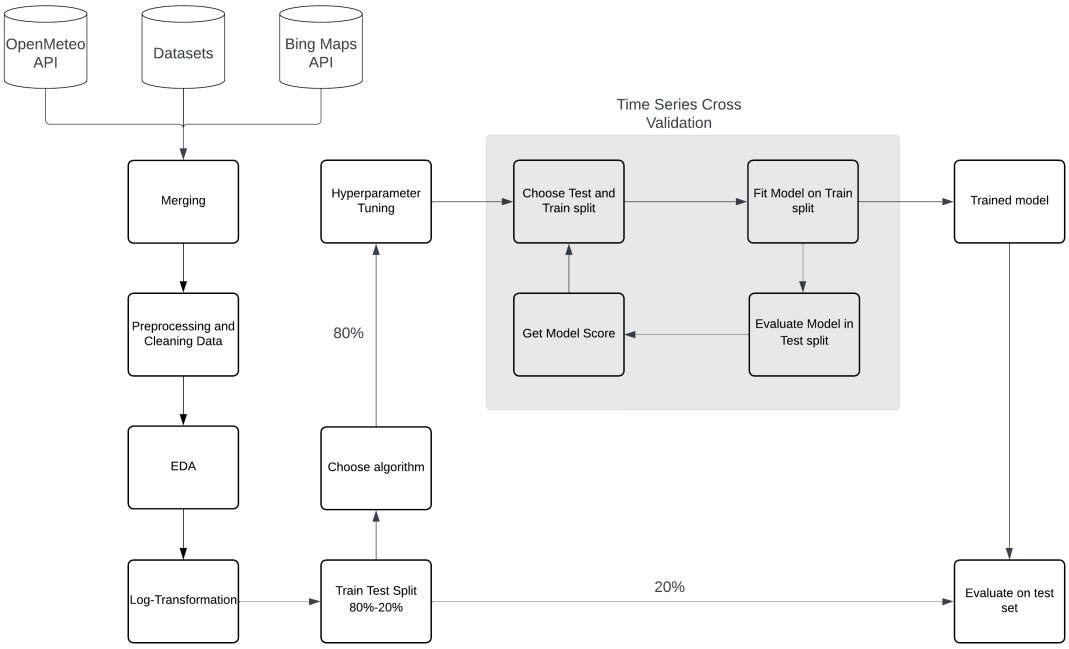

Figure 1: Simplified version of the data pipeline.

3.1 *Dataset Description*

The data used in this study was obtained from multiple sources: the Dutch police, municipalities, Bing Maps API, and weather data services. All of them were publicly available throughout this study, and a combination of them is used to predict the crime rates in Dutch municipalities. The data provided by the Dutch police contains numbers on the total amount of registered crimes per month in each of the 342 municipalities. These crimes are divided into 58 different types of categories, ranging from pick-pocketing to human trafficking and murder; this data is openly available through their portal. The size of this dataset is substantial since it includes

ten years of monthly data for each of the Dutch municipalities. The annual demographic and socio-economic data on municipalities was obtained through the portal of Jive (the host for the municipal data). From this database, variables that showed potential predictive value in relation are selected and downloaded from the portal. Bing Maps API was the source for geographic coordinates for each municipality. This is a tool provided by Microsoft, which makes it possible to connect a longitude and latitude to each of the regions. The last data source is the OpenMeteo API to gather weather data for each day between 2012-2022. This daily weather is then averaged per month to come up with the monthly weather dataset. The dataset with times and municipal data is then merged with the weather data on the variables 'date' and 'time'. All data needed is now in one large dataset. And the last step here is to convert the variables obtained from Dutch to English and all commas to dots.

3.2 *Data Preprocessing*

After obtaining the full dataset, the cleaning and preprocessing steps in Python are described in this part of the study. To start, both the datasets from the Dutch police and Jive were loaded into Python. The crimes per municipality were aggregated to get a monthly total per municipality. The two datasets are now merged on the 'region' and 'Gemeenten' variables for the police dataset and Jive, respectively. The dataset we have now contains all years of data for each of the variables from the Jive dataset, so all columns that do not present data of the year the crime took place are dropped, and other unnecessary columns. At this moment, the Bing Maps API is called upon to generate the longitude and latitude for each region. The next step is to use the OpenMeteo API to generate the daily weather data for the municipality of De Bilt (OpenMeteo API does not support the full amount of regions to generate weather data for) since it is a municipality that lies almost perfectly in the middle of The Netherlands. At the end of the step of aggregating the dataset, all names of current Dutch features are converted to English. To conclude this part, all comma's in the dataset are replaced by dots since commas are not used in the Python language for decimal separators. Also, the Dtype of all numeric variables is converted to 'float64', instead of the Dtype 'object'.

3.2.1 *NA values*

The dataset we have currently has some NA values in the variable, which resembles a region's net labor participation percentage. NA values are values of a variable in the dataset that are unavailable. This van be either

due to it being zero or, for example, due to a limitation in data, the value is simply not 'Not Available'. Having NA values can be problematic for the model applied to the dataset. There are two ways of dealing with these; they can be replaced or deleted. Choosing between these and what to fill it with differs per situation. In this study, the NA values are in a column with values corresponding to a percentage of net labor participation in a region. There are 7482 missing values in this column, while the other values are in order. When the missing values are analyzed, it becomes clear that all missing values are in the region 'Schiermonnikoog'. This municipality is an island with less than 1000 inhabitants. Since this is such a small portion of the dataset and Schiermonnikoog only has so few inhabitants and crimes, the decision was made to exclude Schiermonnikoog from the dataset.

3.2.2 *Categorical variables*

Multiple variables in the dataset are categorical. Since machine learning algorithms do not work well on categorical values, these values should be encoded. This study chooses one-hot encoding to convert the categorical values to numeric ones. The reason for choosing one-hot encoding is the fact that this type is very flexible in use and does not present any assumption which could be made about ordinality.

3.2.3 *Skewed variables*

When an exploratory data analysis was conducted on the entire dataset after cleaning it and dealing with missing or unavailable values, some of the variables representing municipalities' demographic or socio-economic figures were skewed in their distribution. Skewness is a sign which means that there is a lack of symmetry in the distribution of a specific variable. A few examples of problems arising from skewed data are the effect of biasing estimates and reducing overall model performance. Negatively skewed means that the data concentration is on the right side of a distribution with some other values on the left side, while positively skewed data is centered on the left side with a 'tail' to the right. When looking at the six variables in figure 2, the positive skew is clearly visible.

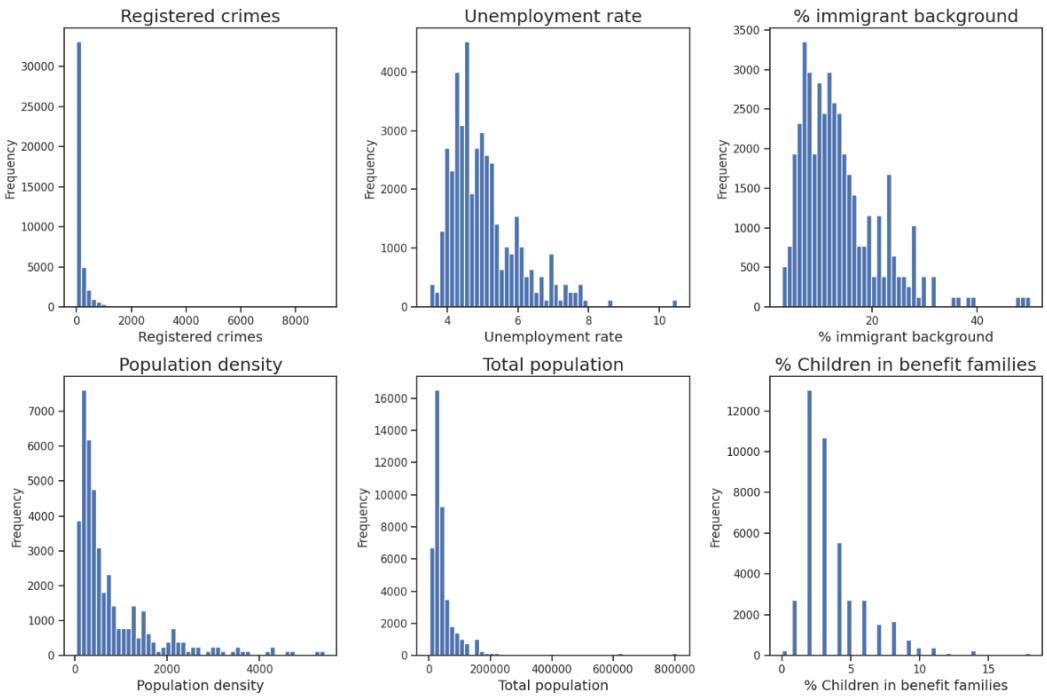

Figure 2: The six variables were found to all be positively skewed.

This skewness limits performance and increases bias in the model, providing us with a reason to transform these variables. There are multiple options to deal with skewness, one of the most popular being log-transformation (Changyong et al., 2014). This applies a logarithmic function to all variables put into the transform. All six variables showing skewness were transformed with a logarithmic function, as seen in figure 3.

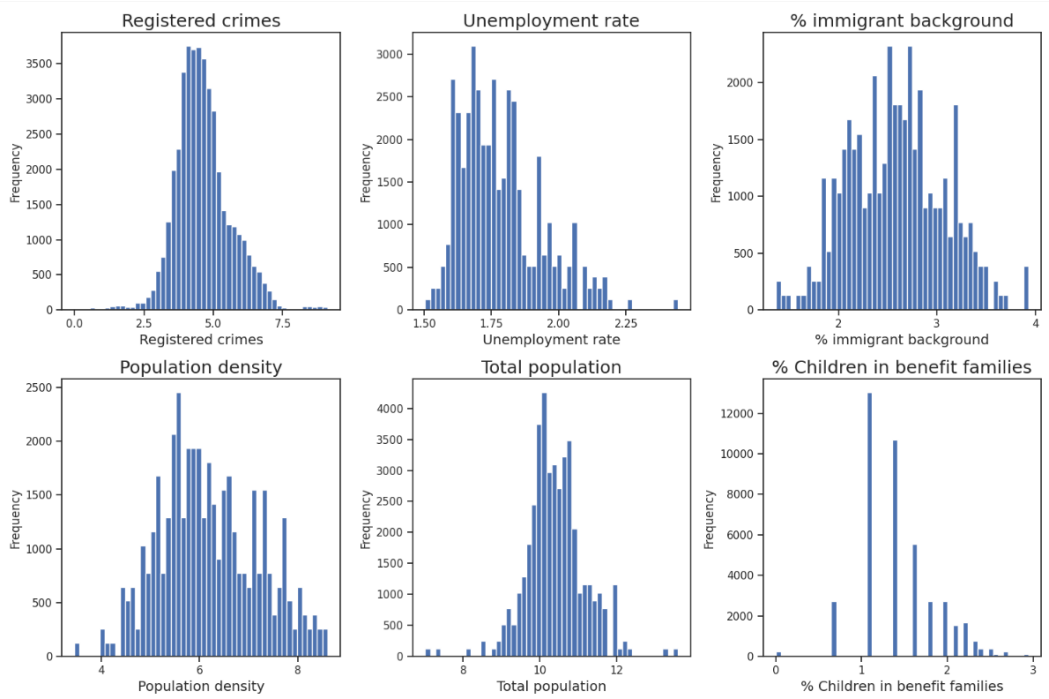

Figure 3: Results after applying a log-transform.

Comparing the distribution of the variables in figure 2 and figure 3, it is clear that all variables are more normally distributed now. After applying the transformation to the data and superficially comparing results between the non-transformed and transformed features after modeling with a linear regresssion, a clear decrease in error and an increase in R-squared was observed.

3.3 *Algorithms*

To answer the main question: *To what extent is it possible to predict future crime statistics in The Netherlands, when using machine learning algorithms?*, multiple algorithms are used to do predictions. The basis for choosing each of the ones used in this research lies in the literature review, which assessed past research and performance. The algorithms chosen for prediction are Decision Tree, Random Forest, XGBoost, KNN, and Multilayer Perceptron. To draw better conclusions in terms of performance, a baseline algorithm is used, in this case, Linear Regression.

The pre-processed dataset is split into two sets with a train-test split. This split is necessary to check the performance of the models. The idea here is that the whole dataset is divided randomly into two parts, one 80% and the other 20%. The 80% will be used to train the models to make

predictions. While the 20% will be used as new data for the model to see how well it performs when it receives data it has not seen before.

3.3.1 *Linear Regression*

The choice for linear regression to be the baseline for conclusions is not random. Linear regression was used in multiple studies mentioned earlier in this study and is often used in machine learning research studies for comparisons. There are multiple reasons why linear regression is a good choice for a baseline to compare results to. Linear regression is a very basic model, making it easy to understand. Compared to the other algorithms suggested in past studies, linear regression is the most inferior type. Also, since it is such a basic model, it is fast to train on large datasets.

Linear regression is a form of a statistical model that can be used to determine the relationship between 2 variables. A linear regression algorithm outputs a line best fitting through all data points. The line which is put out is straight represented in its simplest form by the equation: $y = a + bx$, where a represents the value of y when $x = 0$. b is the change in y per unit of change in x.

3.3.2 *Random Forest*

The Random Forest machine learning algorithm is a decision tree-based ensemble model introduced by Breiman (2001). A Random Forest constructs multiple decision trees when it is being trained and puts out the mean prediction of all these trees. The 'Random' stands for the randomness that is introduced by two steps in the algorithm. Firstly, the data used by each of the trees is bootstrapped, which means that each tree is trained on a different sample of the dataset. This data is selected randomly and replaced each time, meaning some observations could be sampled more than others. Bootstrapping helps to reduce the correlation between trees, resulting in a reduction of variance and improved generalization. Secondly, a randomly selected subset of features is considered for the split. The combination of randomly selecting data and features reduces the correlation between trees even more. Some hyperparameters can be tuned to boost precision or make the model less complex and reduce the computing power needed for the model to run.

- *n_estimators*, the number of trees. Increasing this can improve accuracy but also increases the computational power needed.
- *criterion*, the function which measures the quality of the splits. *'gini'* is a common option for Gini impurity and *'entropy'* is often used to measure information gain.

- *min_samples_split*, minimal number of samples required per leaf node. Increasing the number of samples is a way to reduce overfitting in a model.
- *max_depth*, maximum depth of the trees. Decreasing the maximum depth could also be a way of reducing overfitting in a model.
- *max_features*, determines the maximum number of features to use for finding the best split. Decreasing this parameter results in more randomness and less correlation

3.3.3 *XGBoost*

XGBoost, or *eXtremeGradientBoosting* is, as the name suggests, a particular form of a Gradient Boosting Machine (GBM). Gradient Boosting is, just like Random Forest, a machine learning algorithm based on an ensemble of trees (T. Chen & Guestrin, 2016). Gradient Boosting works by adding decision trees to the ensemble of trees; each tree is trained to predict the residual error from the previous trees. This is then summed up with all calculated residual errors from the trees in the ensemble, resulting in the final prediction. Gradient Boosting trains each newly added tree on the mistakes made by previous trees, thus increasing the performance of the whole model. The process of adding trees to the ensemble has two advantages. It improves the generalization performance of a model, and it reduces bias. The difference between an XGBoost and a Gradient boost is that XGBoost uses a regularization technique, which helps reduce overfitting and the complexity of the model Bentéjac et al. (2021). Some of the hyperparameters which are commonly used and tuned in an XGBoost algorithm are:

- *n_estimators*, the number of trees. Increasing this can improve accuracy but also increases the computational power needed.
- *max_depth*, maximum depth of the trees. Decreasing the maximum depth could also be a way of reducing overfitting in a model.
- *learning_rate*, shrinkage per step size, used to prevent overfitting. If lower, overfitting can be reduced through increasing the number of trees and making the model more computationally heavy.
- *subsample*, the portion of observations which will be randomly sampled per tree. Lower values could decrease overfitting but may increase bias.

XGBoost has performed well in other studies aimed at predicting crime. For example, the studies by Safat et al. (2021) and Zhang et al. (2022) found XGBoost to perform very well on their datasets.

3.3.4 *KNN*

The K-Nearest Neighbors (KNN) algorithm is popular for regression tasks. Unlike linear regression or SVM algorithms, it is a non-parametric type of regression. KNN utilizes distances between instances to predict target variables. This target variable is predicted by finding the K nearest points in the data, which are then averaged on their target values. The 'K' in KNN is linked to the number of neighbors used to get the average from (Ahmed et al., 2010). An advantage of the KNN regression is that it does not make any assumptions about the distribution of data, meaning it can follow non-linear patterns in data (Kohli et al., 2021). A drawback of using KNN regression is the sensitivity to the choice of 'K', setting this parameter too low or too high might result in poor performance for the entire model. Examples of hyperparameters often tuned to increase accuracy include:

- *n_neighbors*, the 'K' which stands for the number of neighbors to consider when predicting
- *weights*, chooses which type of weight function is used in the algorithm. If $'uniform'$ is chosen, it will weigh each neighbor equally. If $'distance'$ is selected, this will assign weights to each neighbor, which is proportional to the inverse of the distance.

In other studies, KNN showed its capability of handling large amounts of data and accurately predicting future crime statistics. Tamilarasi and Rani (2020) used it on a large dataset and concluded that KNN was far superior to a decision tree. Safat et al. (2021) and Malathi and Baboo (2011) also expressed their research to be well complemented by using KNN.

3.3.5 *Mulitlayer Perceptron*

The last model reviewed is a Multilayer Perceptron (MLP), also called a Feedforward Neural network. MLP is an artificial neural network with multiple layers of connected neurons (Murtagh, 1991). An MLP consists of at least three layers, one input, one output, and at least one hidden layer between input and output. An MLP has multiple hyperparameters which can be tuned; the most common ones are:

- number of hidden layers, more layers may allow the model to find more complex relations between variables. Adding more layers does, however, increase the chances of overfitting.
- *units*, is the parameter that specifies the number of neurons in each layer. Selecting too many may result in overfitting, while too few neurons may lead to underfitting.

- *activation*, these functions introduce non-linearity to the neural network. A common choice is, for example *ReLU* (Rectified Linear Unit).
- *opimizer*, the algorithm used for assigning weight in the model during training. For example *adam*, *RMSprop*, or *adagrad*.
- *batch_size*, this is per iteration, how many samples are used when training. Choosing a small batch size can lead to better regularization due to increasing randomness. However, increasing the batch size may result in better noise reduction from outliers
- *epochs*, how often the data is passed through the network built in the model. When this is done many times, generalization is better, but overfitting may occur. While selecting a low amount of *epochs* may have the benefit of faster computing times.

Multiplayer Perceptron was found to perform well in the studies of Safat et al. (2021) and Gonzalez and Leboulluec (2019), both taking place in the United States and both using socio-economic data to predict the amount of crime per region.

3.4 *Hyperparameter tuning*

To realize better results from all models, many of the hyperparameters mentioned before can be altered to fit better for the purpose of this study. These parameters are tuned using a technique called 'Grid Search', which is commonly used in machine learning tasks to find the best-performing combination of parameters to be altered in a model (Paper, 2020). Grid search can smoothly be applied to a model by inputting all possible parameters to check. The working of a grid search is that it tries all combinations of the parameters put into it and evaluates each of the outcomes of the combinations by looking at a metric set in advance. In this case, this metric is the mean absolute error, reasons for this are given in paragraph 3.7. The drawback of Grid search lies in its strength, since it attempts all different combinations of parameters, it can be computationally expensive.

3.5 *Cross-Validation*

Cross-validation is an essential part of validating the results of a model. This technique is used to assess a model's performance while considering its generalization ability as well (Raschka, 2018). A cross-validation involves making splits in the data, to make it possible to have training and

evaluation on different subsets. Doing this on multiple subsets has a few advantages, the main advantage being the more robust model performance evaluation. Training and testing on multiple subsets increase assessment accuracy and can reduce the possibility of a significant impact of variability in the data.

This study applies a Time Series cross-validation to the data, which is split into 5. The steps for this specific type of cross-validation are as follows:

1. *Split the data*, depending on how many splits there are, the data is divided into this amount, which is random, does not overlap, and is of similar size. Figure 4 shows a sketch of what a 5-fold cross-validation looks like in terms of distribution.

2. *Train and evaluate*, the model is trained on all data up to the data that is used as test data. The test data is used to evaluate the model. Both training sets and validation sets thus change for each iteration.

3. *Calculation of performance*, the evaluation metric is averaged over all the folds, resulting in the estimated overall performance of a model.

4. *Hyperparameter tuning*, comparing the outcomes of different inputs of hyperparameters can help to make better decisions and improve models.

In this study, there is chosen to use a 5-fold cross-validation. Firstly, because of the sufficiency of training data, this is 80% of the data each fold contains. Secondly, for its computational cost, which is not as high as, for example, a 10-fold cross-validation. The variance is reduced by a 5-fold cross-validation compared to a single train-test split since it averages the results from all folds.

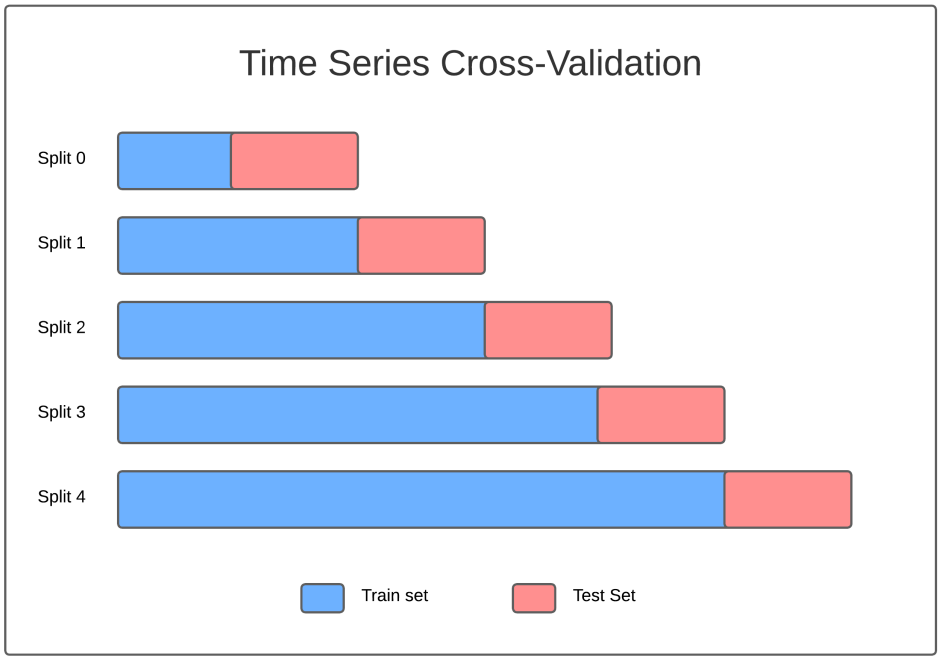

Figure 4: Sketch of a Time Series cross-validation, using 5 iterations.

3.6 *Feature Importance*

To measure the feature importance of the best-performing model, feature importance values and SHAP-values are used. Feature importance values are the scores assigned to each feature to highlight how much impact they made in doing the predictions. SHAP stands for 'SHapley Additive exPlanations', a method used to assign values to features corresponding to how important the features were for the model. SHAP expresses the importance of all input features into numbers and graphs. The main idea of SHAP is that it finds the average contribution of a feature to the outcome of the model's prediction by seeing what difference it makes when the feature is or is not added to the set of features (Mokhtari et al., 2019).

3.7 *Performance Metrics*

To interpret a model's outcome and performance, certain metrics need to be defined in advance. There is a wide variety of metrics used in research to explain the performance of models. It is, however difficult to compare this study with others if both data and models are not comparable, so these evaluation metrics are primarily used to compare the models in this thesis

3.7.1 *RMSE*

Root Mean Squared Error is found by taking the square root of the mean of squared differences of predicted and the real values of a model. RMSE is a popular type of metric since it gives more weight to larger errors than other types of error metrics. While the resulting overall mean error might be low for a certain model, there is a chance of significant wrong predictions. To be able to take these into account, RMSE is a viable option. Larger errors can have a serious impact on the quality of a prediction. The RMSE is a common metric to take these large errors into account and performs well doing this (Chai & Draxler, 2014). Penalizing errors can be important in scenarios where you, for example want to know if the spread of errors is big compared to a high average error (MAE).

3.7.2 *MAE*

Another metric commonly used for the performance evaluation of regression models is the Mean Absolute Error. The MAE measures the mean of differences between the predicted values by the model and the real values in absolute numbers. Other than RMSE, MAE does not alter the weights of different sizes of errors. This means that the MAE presents an average error, allowing us to determine how well the model does in general. The mean average error allows for a straightforward assessment of the model's performance being evaluated. MAE is also reasonably robust against outliers and large deviations, making it appropriate for the type of predictions in this study. For these reasons, the MAE is chosen as the main evaluation metric for this study.

3.7.3 *R-Squared*

RMSE and MAE are metrics used to measure the error of prediction a model has. R-squared (R^2) is used to measure how much of all variance is explained by the predictive model. The value R-squared produces is between 0 and 1, this number can be considered to be similar to the percentage of variance of the target variable explained by modeling the independent variables. An R-squared can, in theory, be negative when a model performs poorly when evaluated. Chicco et al. (2021) argued that R-squared is a good option when evaluating regression models compared to other measures for model fit. R-squared is also the metric used by many other similar studies in the field of crime prediction (for example (Aziz et al., 2022) and (V. Kumar, 2023)), which makes it easier to compare the performance of studies from different regions in the world.

3.8 *Programming language and external applications*

This study was conducted in Python (Van Rossum and Drake (2009), version 3.10.11) programming language, which has a wide variety of applications. Python was the language of choice for this study because of its versatility, user-friendliness, and a vast collection of libraries for data analysis, manipulation, evaluation, and visualization. For this study, the *sklearn* library is used for a significant portion of the work in Python. *Sklearn* (Scikit-learn) is a machine learning library in Python, which consists of many useful tools for all stages of data analysis. It offers powerful instruments for preprocessing, feature selection, model predictions, and evaluation of those models. *Sklearn* also works well with other popular libraries such as *Numpy* and *Matplotlib*, which is also used during this study. For the MLP model, the *Tensorflow* library was utilized. *Tensorflow*

The geographical coordinates were obtained using the Bing Maps API, which provides open access to its resources and can be called upon through code in Python. This allows this study to add geographical locations to all municipalities shown in the data. The weather data was obtained through OpenMeteo, which is open to all users and provides data on various places worldwide. The municipalities' geographical locations can also be used to gather weather data from the OpenMeteo API.

4 RESULTS

The results from all models are presented in the chapter. The results will also be discussed shortly per the algorithm. Table 1 provides all results of the evaluation metrics when each of the trained models was used to predict the values of the test set, which was set apart before the training. Since XGBoost clearly performed the best in the overall evaluation metrics, this is our best model. The performance of the XGBoost model will be evaluated later in this chapter, identifying its most important features in section 4.6 and analyzing its errors in section 4.7.

Test set	Evaluation Metrics		
Models	RMSE	MAE	R^2
Linear Regression	98.439	33.392	0.963
Random Forest	43.869	18.174	0.991
XGBoost Regression	34.526	16.318	0.995
KNN	41.526	19.134	0.993
Multilayer Perceptron	48.898	22.096	0.990

Table 1: Performance results per algorithm, predictions of total crimes per municipality, when evaluated on the test set.

4.1 *linear Regression*

The linear regression algorithm was used in this study as a baseline to compare other algorithms. This is a basic kind of regression, which makes it a good measure for comparison. Linear regression was, after feature engineering, the least predictive model for crime prediction using our data on both the validation and test sets. The root means squared error of 98.439 is high when considering that the mean of the registered crimes variable is 194.23. The mean absolute error is better with 33.392, although still indicating significant errors in prediction, even though this result is only to be able to set a baseline for comparison. The R-squared is, especially compared to the other models, quite high with approximately 96.3% of variance explained by the linear regression model.

4.2 *Random Forest*

The results from the Random Forest regression were calculated using the Gridsearch outcome. These outcomes suggested a maximum depth of 15 trees and a minimum sample split of 5 and 150 trees to do predictions with. The results after a time series cross-validation with 5 splits on the test set were a root mean squared error of 43.869, a mean absolute error of 18.174, and an R-squared of 0.991. The Random Forest has the second-best performance when comparing our primary evaluation metric (MAE). The high R-squared indicates a good performance of variance explained by the model, whereas the low RMSE and MAE correspond to relatively low errors in prediction.

Parameter	Tested	Best
Max depth	5, 10, 15	15
Min sample split	2, 5, 10	10
# of estimators	100, 200, 300	150

Table 2: Optimal hyperparameters for Random Forest

4.3 *XGBoost*

Outcomes of the Gridsearch indicated using a learning rate of 0.1, maximum depth of 10, and 100 trees when doing the 5-fold cross-validation on the XGBoost model. When applied to the dataset, the outcomes of this combination of hyperparameters were the best when compared to the other model used. The XGBoost algorithm best explained the variance of the predicted crimes with an R-squared of 0.995. The errors were also the lowest of all models, having an RMSE of 34.526 and an MAE of 16.318. Also, before tuning the hyperparameters, the XGBoost algorithm showed its capabilities by having good scores in all metrics.

Parameter	Tested	Best
Learning rate	0.1, 0.01, 0.001	0.1
Max depth	3, 5, 7	5
# of estimators	100, 200, 300	300

Table 3: Optimal hyperparameters for XGBoost

4.4 *KNN*

When reviewing the literature, K-nearest neighbors was close to being the most popular choice in crime predictions. The results of applying the algorithm to the data justify this popular choice. With an RMSE of 41.526 and an MAE of 19.134, KNN does not outperform Random Forest when compared to the MAE, but the RMSE is lower, suggesting fewer extreme errors than Random Forest.

Parameter	Tested	Best
# of neighbors	3, 5, 7, 9	5
Weights	uniform, distance	uniform

Table 4: Optimal hyperparameters for KNN

4.5 *Multilayer Perceptron*

The achieved results of the MLP were much higher after hyperparameter tuning, resulting in an RMSE of 48.989, an MAE of 22.096, and an R-squared of 0.990. The Multilayer Perceptron in this study consists of three hidden layers, with 64, 32, and 16 units, respectively, combining this with the 'ReLU' activation function and the 'Adam' optimizer.

Parameter	Tested	Best
Batch size	16, 32	32
Epochs	50, 100	100

Table 5: Optimal hyperparameters for Multilayer Perceptron

4.6 *Feature importance*

As mentioned before, identifying the most important features gives insight into what regional factors could be used by law enforcement to distribute resources and which variables would be most interesting to investigate further in other research. To review the importance of the predictive features utilized in this study, their overall feature importance is evaluated. This is done by calculating and visualizing the SHAP values of all features used in this study. The feature importances of all features were also calculated with the built-in 'Scikit-learn' tools and can be found in appendix 6.

The plot of the SHAP values in figure 5 provides a calculation and visualization of the impact of each of the XGBoost model's features. Note that the feature of the total population has been left out to make the other features better interpretable (the SHAP summary plot with the total population included can be found in appendix 7). Firstly, The output of a SHAP summary plot is ordered by the highest SHAP value, in this case, the feature representing the year. Secondly, the SHAP value can be positive or negative, referring to what it does to the target variable (meaning if it increases or decreases the predicted amount of crimes). Thirdly, the color blue means a low value of a feature, and red a high value.

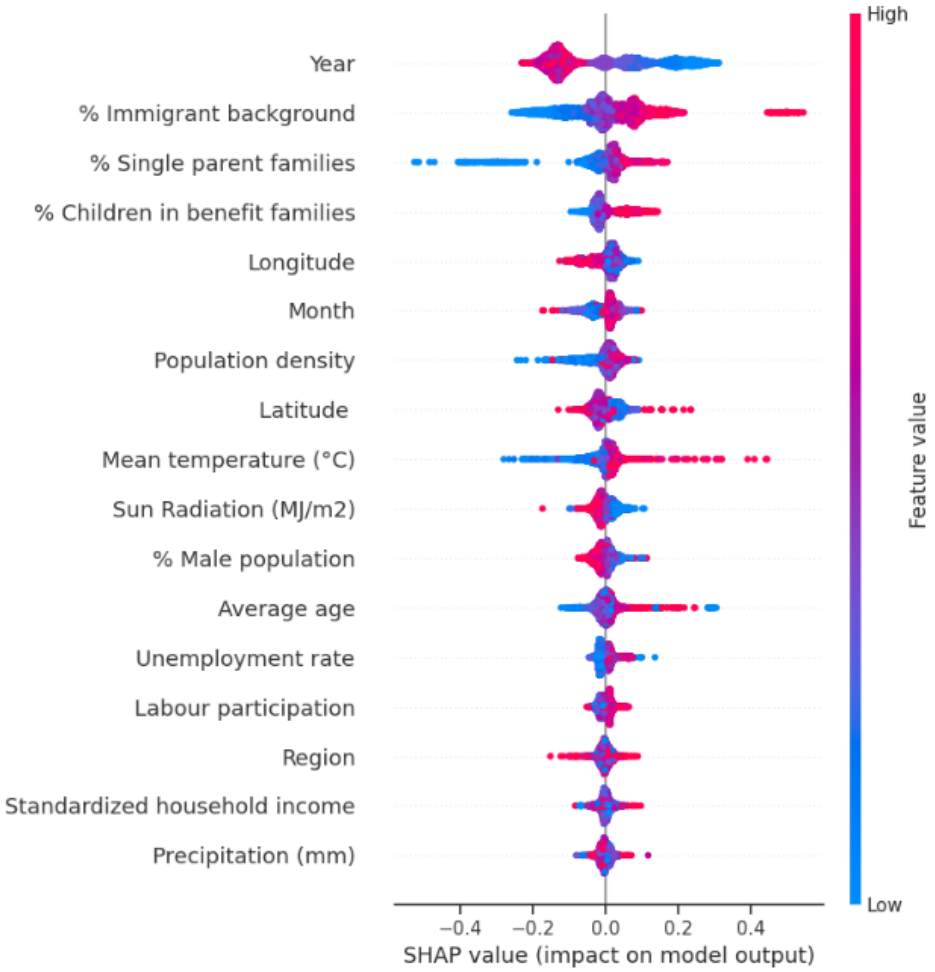

Figure 5: SHAP values, having excluded the total population feature (see appendix 7 for the total population feature).

In the plot, we can see several results. First of all, a higher number of years impacts the model with negative SHAP values, meaning that the model's prediction decreases if the years increase. Secondly, a higher percentage of people with an immigrant background in a municipality tends to increase the number of predicted crimes by the model. The same conclusion can be made for both the feature of single-parent families and children in benefit families. The mean temperature feature also positively influences the number of crimes predicted when the value increases, suggesting the model finds a relation between warm weather and a higher number of crimes.

Other interesting finds are the absence of high SHAP values for features like labor participation, income, and unemployment rate. Although these features showed potential when reviewing the literature, they did not stand

out in prediction through the XGBoost model reviewed here. On the other hand, mean temperature and sun radiation were reasonably influential in the model's predictions, even though little was known about their effect on crime predictive models. The feature representing the total population is by far the most influential in the model, as seen in the figure in appendix 7.

4.7 *Model performance*

To evaluate the performance of the best-performing model, an analysis is done to review its errors. The analysis involves finding the relation between population and the errors made in the model's prediction of crimes in the municipality. The two cities with a population of over 400,000 people were excluded from this analysis to have a more detailed focus on all other predictions. The plot in figure 6 provides a scatter plot containing the errors in the plot, which represent the differences between the actual values and those predicted by the XGBoost model. On the sides of the scatter plot, bar charts depict the distribution of instances with either a certain error or a municipality with a certain population.

When looking at the figure, the error distribution seems to be related to the size of the population, meaning larger populations have higher errors. This is not unexpected when considering that the total population feature is by far the most important feature in the predictive model. We also know a strong relationship exists between the population size and the number of crimes committed in the municipality. Although this plot seems to indicate that the errors were very spread, this assumption can be tempered when we look at the distribution of the instances with a certain error in population size. The error distribution depicts the relatively dense concentration of errors around a residual error of 0. At the same time, the distribution of population size lies mainly between 0 and 50,000.

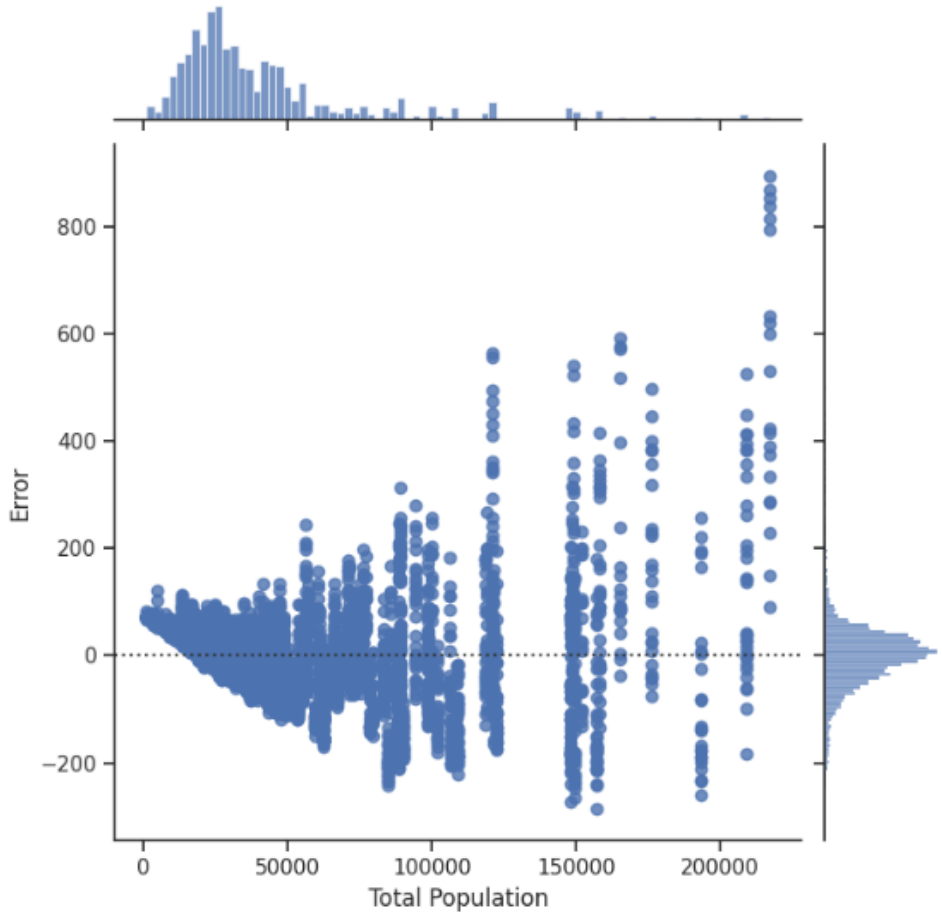

Figure 6: Prediction error versus the total population (of municipalities with a population of less than 400,000).

5 DISCUSSION

This study aimed to determine to what extent it is possible to predict future crime statistics in The Netherlands using machine learning algorithms. Through its course, many past studies were brought forward. The algorithms these studies found helpful were investigated and considered for crime prediction in The Netherlands. The algorithms most likely performing well in The Netherlands were selected to answer the main research question.

5.1 *Results*

It is only possible to answer the research questions by looking at the results from the models. These models were all applied to the same dataset, and each of their hyperparameters was tuned as much as it was computationally possible at the time of research. The results obtained from modeling the data using the algorithms give a clear view of which models were weak and which had useful predictions. The tree-based algorithms, in this case, XGBoost and Random Forest, got great results in all evaluation metrics. KNN performed almost equally well as Random Forest, but its MAE was slightly higher. Multilayer Perceptron scored the worst regarding the error and R-squared after the baseline Linear Regression.

The XGBoost model proved to be the most accurate in predicting crime in The Netherlands, with a mean absolute error of 16.318. The RMSE stood at 34.526, and the R-squared value was an impressive 0.995. These values indicate that XGBoost would be the most viable choice out of the evaluated models when predicting crimes in the Netherlands with a similar dataset. This result that tree-based algorithms would perform well is in line with studies from the literature review and the study conducted by Zhang et al. (2022) in particular. Using the historical data of crimes and weather, in combination with demographic and socio-economic features, the model was able to calculate reasonably accurate predictions. However, it is difficult to compare these results since not many studies have been done on this subject using the same features in The Netherlands.

The features used in this study were evaluated on their performance and impact. The most important results from this analysis were the importance of the features representing the year, the population with an immigrant background, and single-parent families. Notably, the temperature positively influences the number of predicted crimes. When evaluating model performance, the relation between population size and error spread is immediately clear for the model. Still, this spread can be misleading since most combinations of errors and population sizes are concentrated instead of highly spread.

5.2 *Limitations*

This study has made many decisions, and certain limitations must be acknowledged. Firstly, the data put into all models was not as extensive as, for example, in studies by Zhang et al. (2020) and Safat et al. (2021). The crime data from the police only contained monthly numbers, while data such as the time, exact location within a municipality, and age of people involved were absent. For example, the data obtained from the

municipalities was all annual since that was the only publicly available data. The weather data obtained from OpenMeteo was national instead of unique per municipality. The reason for only using national weather data was that the OpenMeteo API only allowed for a maximum number of requests per key. The absence of extensive data on each crime committed results in a lower accuracy and higher error for all models trained on that data. While Linear Regression, KNN, Random Forest, XGBoost, and Multilayer Perceptron are well-known algorithms, these only represent a small selection of algorithms that can be used for regression tasks like crime prediction. Researching other, more advanced types of algorithms for modeling has the potential to increase the accuracy of the models.

5.3 *Future Research*

Even though this study is almost concluded, research into crime prediction is not. When doing crime predictions in The Netherlands, this study found some suggestions to explore and incorporate in future research. First, data aggregation would be at the centre of any new research into crime prediction in The Netherlands. For better predictions, obtaining as much unique data per instance of crime is necessary to minimize errors and maximize performance. This could be specific to the crime or the municipality in which it was committed. Secondly, try other algorithms, such as Long Short-Term Memory, an artificial neural network used by Safat et al. (2021). LSTM can capture relations in sequential data better than other variants like MLP. Also, SVM was an algorithm that showed potential in the studies of Shah et al. (2021) and would be interesting to apply in The Netherlands. Using these suggestions in future research might help law enforcement reduce crime and feelings of unsafety among citizens Johansson and Haandrikman (2021) and Visser et al. (2013). In this study, there was no normalization of the target variable. It is worth noting that normalizing the amount of crime by population allows for a more fair comparison of regions and could increase the model's performance (Wang et al., 2016).

6 CONCLUSION

This research study attempted to contribute to the growing demand for more accurate predictions in crime statistics in the Netherlands. The main question of this research was: *To what extent is it possible to predict future crime statistics in The Netherlands when using machine learning algorithms?*. To be able to answer this main question, research questions were covered using literary research and modeling.

- *How well do Random Forest, XGBoost, KNN, and Multilayer Perceptron perform when predicting crime?*

  After modeling and testing all machine learning algorithms, it became clear that there were substantial differences in their ability to use the data supplied and do accurate predictions on a different test set. The baseline from a linear regression algorithm was a low standard, thus outperforming all algorithms researched. Multilayer Perceptron did much better than the baseline, even reducing the primary evaluation metric MAE by 33%. KNN scored better than Random Forest in evaluation metrics other than the MAE, but its mean absolute error was slightly higher than what Random Forest was able to produce. To conclude, XGBoost was the best-performing model in all metrics with an MAE of 16.318, an RMSE of 34.526, and an R-squared of 0.995

- *Which features are most important for the best-performing models when predicting crime?*

  After a municipality's population was found to be the most important feature by far, other features also significantly impacted the model's predictions. After analyzing these features using their SHAP values, it became clear that the features depicting the percentage of people with an immigrant background and the percentage of single-parent families impacted the model the most, followed by population density numbers and the percentage of children living in benefit families.

- *How does the best-performing model differ in performance when population size increases?*

  A residual plot with the errors of the XGBoost model was created to answer this research question. This plot is combined with two histograms on each axis for a more informed view of the residuals of the XGBoost model. The errors do increase when the population size increases, which is to be expected when you consider that population size is the most essential feature in the model. At first glance, the distribution also looks very spread out, but when the histograms are taken into account, these conclusions are softened.

To answer the main question of this study, using annual socio-economic and monthly weather features, it is possible to predict the amount of crime in The Netherlands with a mean absolute error of 16.318 for the XGBoost algorithm.

7 SOURCE/CODE/ETHICS/TECHNOLOGY STATEMENT EXAMPLE

Data Source: The data used in this thesis was obtained through various sources. Crime statistics were freely downloaded from the Dutch police database, while all variables related to municipalities were obtained from the 'Waarstaatjegemeente' database (Jive), also freely available online. The weather data used in the study was obtained from the Open-Meteo API, which provided free access to the required information. Bing Maps API was furthermore used to aggregate the geographical locations of municipalities, also free for use. The author acknowledges the original sources of the data and code used in this thesis. The images displayed in this study were all the work of the author. This thesis was written in Overleaf, and Grammarly was utilized to do grammar checks. No data was collected from human participants or animals for this study.

REFERENCES

Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric reviews*, *29*(5-6), 594–621.

Algahtany, M., Kumar, L., & Barclay, E. (2022). A tested method for assessing and predicting weather-crime associations. *Environmental Science and Pollution Research*, *29*(49), 75013–75030.

Alves, L. G., Ribeiro, H. V., & Rodrigues, F. A. (2018). Crime prediction through urban metrics and statistical learning. *Physica A: Statistical Mechanics and its Applications*, *505*, 435–443.

Archer, J. (2022). *Male violence*. Taylor & Francis.

Aziz, R. M., Hussain, A., Sharma, P., & Kumar, P. (2022). Machine learning-based soft computing regression analysis approach for crime data prediction. *Karb Int J Mod Sci*, *8*(1), 1–19.

Bentéjac, C., Csörgő, A., & Martınez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*, 1937–1967.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Cavadas, B., Branco, P., & Pereira, S. (2015). Crime prediction using regression and resources optimization. *Progress in Artificial Intelligence: 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings 17*, 513–524.

Chai, T., & Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, *7*(3), 1247–1250.

Changyong, F., Hongyue, W., Naiji, L., Tian, C., Hua, H., Ying, L., et al. (2014). Log-transformation and its implications for data analysis. *Shanghai archives of psychiatry*, *26*(2), 105.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Chen, X., Cho, Y., & Jang, S. Y. (2015). Crime prediction using twitter sentiment and weather. *2015 systems and information engineering design symposium*, 63–68.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, *7*, e623.

Cortoni, F., Babchishin, K. M., & Rat, C. (2017). The proportion of sexual offenders who are female is higher than thought: A meta-analysis. *Criminal Justice and Behavior*, *44*(2), 145–162.

Goel, S., Rao, J. M., & Shroff, R. (2016). Precinct or prejudice? understanding racial disparities in new york city's stop-and-frisk policy.

Gonzalez, J. J., & Leboulluec, A. (2019). Crime prediction and socio-demographic factors: A comparative study of machine learning regression-based algorithms. *Journal of Applied Computer Science & Mathematics*, *13*(27).

Jenga, K., Catal, C., & Kar, G. (2023). Machine learning in crime prediction. *Journal of Ambient Intelligence and Humanized Computing*, *14*(3), 2887–2913.

Johansson, S., & Haandrikman, K. (2021). Gendered fear of crime in the urban context: A comparative multilevel study of women's and men's fear of crime. *Journal of Urban Affairs*, 1–27.

Kassem, M., Ali, A., & Audi, M. (2019). Unemployment rate, population density and crime rate in punjab (pakistan): An empirical analysis. *Bulletin of Business and Economics (BBE)*, *8*(2), 92–104.

Kohli, S., Godwin, G. T., & Urolagin, S. (2021). Sales prediction using linear and knn regression. *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*, 321–329.

Kubrin, C. E., & Ishizawa, H. (2012). Why some immigrant neighborhoods are safer than others: Divergent findings from los angeles and chicago. *The Annals of the American Academy of Political and Social Science*, *641*(1), 148–173.

Kumar, A., Verma, A., Shinde, G., Sukhdeve, Y., & Lal, N. (2020). Crime prediction using k-nearest neighboring algorithm. *2020 International conference on emerging trends in information technology and engineering (IC-ETITE)*, 1–4.

Kumar, V. (2023). Crime data analysis using machine learning models. *Applied Technologies: 4th International Conference, ICAT 2022, Quito, Ecuador, November 23–25, 2022, Revised Selected Papers, Part I*, 296–309.

Levine, E. S., Tisch, J., Tasso, A., & Joy, M. (2017). The new york city police department's domain awareness system. *Interfaces*, *47*(1), 70–84.

MacDonald, J. M., Hipp, J. R., & Gill, C. (2013). The effects of immigrant concentration on changes in neighborhood crime rates. *Journal of Quantitative Criminology*, *29*, 191–215.

Malathi, A., & Baboo, S. S. (2011). Enhanced algorithms to identify change in crime patterns. *International Journal of Combinatorial Optimization Problems and Informatics*, *2*(3), 32–38.

Meijer, A., & Wessels, M. (2019). Predictive policing: Review of benefits and drawbacks. *International Journal of Public Administration*, *42*(12), 1031–1039.

Mokhtari, K. E., Higdon, B. P., & Başar, A. (2019). Interpreting financial time series with shap values. *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, 166–172.

Murtagh, F. (1991). Multilayer perceptrons for classification and regression. *Neurocomputing*, *2*(5-6), 183–197.

Paper, D. (2020). Scikit-learn regression tuning. *Hands-on Scikit-Learn for Machine Learning Applications: Data Science Fundamentals with Python*, 189–213.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.

Safat, W., Asghar, S., & Gillani, S. A. (2021). Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques. *IEEE Access*, *9*, 70080–70094.

Shah, N., Bhagat, N., & Shah, M. (2021). Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention. *Visual Computing for Industry, Biomedicine, and Art*, *4*, 1–14.

Shamsuddin, N. H. M., Ali, N. A., & Alwee, R. (2017). An overview on crime prediction methods. *2017 6th ICT International Student Project Conference (ICT-ISPC)*, 1–5.

Short, M. B., D'orsogna, M. R., Pasour, V. B., Tita, G. E., Brantingham, P. J., Bertozzi, A. L., & Chayes, L. B. (2008). A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences*, *18*(suppo1), 1249–1267.

Sommer, A. J., Lee, M., & Bind, M.-A. C. (2018). Comparing apples to apples: An environmental criminology analysis of the effects of

heat and rain on violent crimes in boston. *Palgrave communications*, *4*, 138.

Tamilarasi, P., & Rani, R. U. (2020). Diagnosis of crime rate against women using k-fold cross validation through machine learning. *2020 fourth international conference on computing methodologies and communication (ICCMC)*, 1034–1038.

Van de Veer, E., De Lange, M. A., Van Der Haar, E., & Karremans, J. C. (2012). Feelings of safety: Ironic consequences of police patrolling. *Journal of Applied Social Psychology*, *42*(12), 3114–3125.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

Visser, M., Scholte, M., & Scheepers, P. (2013). Fear of crime and feelings of unsafety in european countries: Macro and micro explanations in cross-national perspective. *The Sociological Quarterly*, *54*(2), 278–301.

Wang, H., Kifer, D., Graif, C., & Li, Z. (2016). Crime rate inference with big data. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 635–644.

Xie, M., & Baumer, E. P. (2019). Neighborhood immigrant concentration and violent crime reporting to the police: A multilevel analysis of data from the national crime victimization survey. *Criminology*, *57*(2), 237–267.

Zhang, X., Liu, L., Lan, M., Song, G., Xiao, L., & Chen, J. (2022). Interpretable machine learning models for crime prediction. *Computers, Environment and Urban Systems*, *94*, 101789.

Zhang, X., Liu, L., Xiao, L., & Ji, J. (2020). Comparison of machine learning algorithms for predicting crime hotspots. *IEEE Access*, *8*, 181302–181310.

APPENDIX A

Feature	Importance
Total Population	0.837480
% Immigrant background	0.048962
% Single parent family	0.034593
% Children in benefit families	0.021219
Year	0.010443
Population density	0.008184
% Male population	0.007853
Unemployment rate	0.005313
Standardized household income	0.004708
Longitude	0.004242
Latitude	0.003466
Average age	0.003190
Labour participation	0.002671
Mean temperature	0.002131
Month	0.001991
Region	0.001827
Sun radiation (MJ/m^2)	0.001050
Precipitation (mm)	0.000677

Table 6: Feature importances for XGBoost, calculated using the 'feature_importances_' tool in Scikit-learn.

APPENDIX B

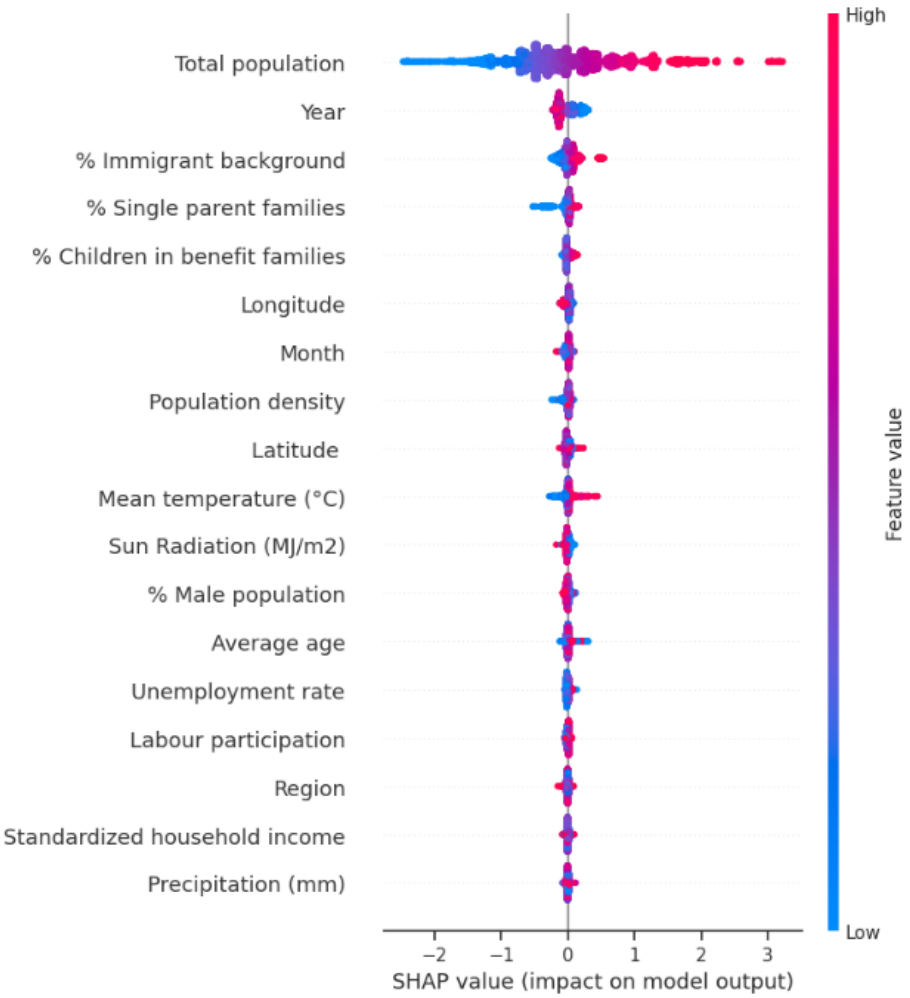

Figure 7: SHAP values, including the total population feature.