NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

BC2407 Analytics II: Advanced Predictive Techniques
Semester 2, AY 2020/21

Reducing Crime Rate: Smarter Decisions During Crime Handling

Prepared By	Chen Gangzhe (U1822840E) Jethro Phuah An Ping (U1921386F) Shannon Tan Xinyi (U1921019B) Tan Yu Xuan Remus (U1911741K) Vohra Ishaan (U1911740H)
Seminar Group	4
Team	2
Tutorial Instructor	Liu Peng
Date of Submission	04/04/2021

Table of Contents

Executive Summary

As cities urbanize and are inhabited by denser populations, law enforcement agencies find it increasingly harder to police and protect their citizens. For police departments that already operate with stretched resources, data analytics offers an innovative approach towards fighting crime. Recently, a growing number of law enforcement agencies rely on these analytics, built on historic crime and offender data, to predict the likelihood of a certain type of crime being committed, and in extension, whether released offenders are likely to commit a certain crime.

Our project aims to construct a predictive model, based on crime in the city of Chicago, IL, to successfully determine where in a city a crime is more likely to be committed based off of two chief datasets- The Chicago Crime (CC) dataset and National Corrections Reporting Programs (NCRP). Over the course of the project, we aim to apply our techniques to propose a data-driven method to predicting crime in Singapore, and, through proper interpretation and execution, lead to a country where crime is better targeted, fought, and where the population is better guarded against threats.

We begin by choosing two datasets and assessing their suitability for our requirements based on the reliability, completeness, impartiality, and usability of data. Concluding that our datasets- public and comprehensive that span years and offer a thorough look of crime rates all over Chicago- meet all these criteria, we begin by cleaning our data for use. This includes excluding missing and null values, irrelevant data such as unique IDs, and exact coordinates of a crime. We scanned for logical errors and processed all dependent variables and dropped any redundant variables. Some continuous variables were scaled so that they may be used for neural networks as part of our analytics framework. Conducting exploratory analyses on our data using Python revealed key insights that we later use as a lead when formulating our approach.

We move on by deciding on a multi-pronged approach to predict the primary type of crime that may occur in the future, as well as the likelihood of a past offender recidivating. For predicting crime, our approach uses two analytical models- CART and a Logistic Regression Model. After running both models, we found that a CART model yields relatively more optimal results, albeit with reasonable limitations. Our recidivism index, on the other hand, was processed using Artificial Neural Networks and Random Forest Classifications- ultimately deciding that here a Neural Network fits best, with reasonable limitations as well.

We then move on to an evaluative reflection on our model- drawing on practical and real-life cues to assess the feasibility and extendibility of our model with special respect to its use for judiciaries and practicality in law.

Drawing parallels between our project and its relevance and ability to be applied to Singapore, considerations were put forward that lay out our projects' potential, current limitations, and recommendations for use in the real world. These aim to guide users of our project to the potential that our project and data analytics possess in solving the problem of urban crime.

1 Introduction of Business Problem

As the global population increases and societies become increasingly interwoven, governments around the world find it increasingly challenging to allocate limited resources to law enforcement efforts and controlling crime in their countries. The usage of modern analytical techniques to aid in crime management is termed as predictive policing. (Bachner, 2013)

Big data and analytics models enable law enforcement agencies to predict criminal activity in local communities, as well as uncover insights and patterns that can aid law enforcement in utilising limited resources more effectively by making use of domestic and foreign crime data collected through past and ongoing crime prevention activities. (Jayaweera et al, 2015)

In this project, we will be exploring the criminal activity in the United States as its crime rate is about average in the whole world. (Crime rate by country, 2021)

1.1 Identifying Type of Crime For Policing Decisions

In most cases, the decision as to who to send to an incident largely falls to the hands of the dispatcher and not the response units. Often, without sufficient information, dispatchers broadcast a blanket call to all units enquiring which officers or teams are in the vicinity and are able to attend to the problem. Hence, much of their decision-making process is done in real time and usually in a hurry. However, this method does not always result in the most efficient response unit being chosen for the task. (Dunnett, S., Ms, Leigh, J., Ms, & Jackson, L., Ms. 2018, February 22)

Additionally, when solving crimes, the allocation of police resources can be viewed as two problems- the first being quantitating policing hours required in a case, and the second being qualitatively assessing the type and how policing units should be deployed (Robert P. Shumate, Richard F. Crowther, 1966). Appropriate allocation of police teams to solve cases and conduct patrol activity is thus essential for more efficient crime handling.

Using data analytics on past criminal record data in the area, dispatchers can better pinpoint the type of crime that will happen. With this information, dispatchers can confidently make more informed decisions to assign suitable resources and response units to crisis calls. This may mean assigning higher priorities to calls that are predicted to be of a violent nature. Cases requiring more time and manpower can also be identified so that resources will be distributed more efficiently.

1.2 Criminal Recidivism

According to Prison Legal News, it is anticipated that 17% of released prisoners in Illinois will commit another offence within one year, and around 43% will reoffend within 3 years. In Chicago 2016, out of 71,551 new convictions, 89% were reoffenders headed back to prison. Criminal recidivism is a significant problem, especially in the United States, where 45% of total prisoners released in 2005 by 30 states were arrested one year after release and 83% were arrested 9 years after release (Clark, 2019). There are many issues arising from high rates of recidivism, such as increased financial burden and decreased level of safety in the public.

In addition, certain types of crime may see higher recidivism rates. In the United States, over 50% of the former prisoners convicted of property crimes were arrested in the first year following their release, followed by 39% of re-releases convicted of violent crimes, 43% with drug convictions and 41% with public order convictions.

Clearly, crime prevention does not just stop at arresting felons and placing them behind bars. Instead, more resources and efforts are needed to reduce criminal recidivism and increase public safety levels. (Lyon, E., Mr. 2019, February 5).

By analysing past criminal records of people, the predictive model can help to determine the potential likeliness of a criminal committing another crime after serving their jail term. With this information, judiciaries may be able to make more informed decisions in terms of the criminal sentence. More attention and resources can also be put towards groups of felons who are identified as being more vulnerable to recidivism so that their chances of re-committing crimes can be minimised.

1.3 Applying Insights to Practice

The primary goal of our project is to predict the type of crime that will happen, and to predict potential recidivous behaviour in released convicts. After the model is able to accurately predict criminals who are more likely to recommit, relevant agencies can narrow down their scope to a more targeted group as they identify, delve into and analyse any factors that can potentially reveal signs of higher propensity for recidivism. This helps authorities better allocate their limited resources and make their efforts in re-integrating ex-convicts back to society more effective. It must be noted, however, that statistical efficiency does not equal true efficiency of a system of law enforcement - it may be argued that a truly efficient system of law enforcement is once that focuses more on systemic changes to eliminate the reasons that drive people to commit crimes in the first place. Our project, however, excludes such arguments and focuses solely on providing authorities with a clearer picture on what types of crime will be committed and the potential tendency of an offender to recommit crimes. The task of analysing the motivations behind the offenders actions falls onto those who may choose to utilise our project in the future.

2 Analytics Solution

2.1 Approach: Smarter Decisions During Crime Handling

In this project, we aim to create analytics models that can predict the following:

1. Types of crime that may occur in the state of Chicago in the future
2. Likelihood of a past offender committing another crime in the future

These models may be interpreted and used by the following:

- Policing agencies to engage in more efficient resource allocation and scheduling.
- Judiciary systems during sentencing.
- Policy makers when evaluating the effectiveness of their penal systems
- Governments when building programs aimed towards helping released prisoners re-enter society.

Together, the models would allow for more informed decisions to be made when handling criminal cases in the hopes of increasing policing efficiency and reducing recidivism.

2.2 Selection of Data

The focus of our project and analysis is on the city of Chicago, in the United States of America. This city was chosen in particular for its high crime rate (City-data.com, n.d), and for its population and metropolitan layout. It is a city with a large amount of published crime reports and data. To predict our output variables, we have made use of two different datasets which are available in the following .csv files:

- Predicting Primary Type of Crime:
    - CC.tsv: Dataset on incidents of crime that occur in the City of Chicago
    - Community Area Names.csv: To map the community area names to their corresponding area code in the CC dataset (for data exploration)

- Predicting Criminal Recidivism:
    - NCRP.csv : Dataset on history of offenders in prison

The Chicago Crime (CC) dataset and National Corrections Reporting Programs (NCRP) and contain about 7 million and 11 million records of data respectively. This gives us a data pool that is large enough to infer patterns from and is sufficiently reliable in terms of its source. Our data source offers crime statistics from as early as 1983 (although for the focus of our report relates to the past 20-30 years), and is currently active and published annually, allowing us to decipher trends at both a macro and micro level.

3 Data Preparation

Data preparation is to be carried out before building the model. We have chosen two sample datasets to perform our analysis. Our first sample dataset is provided by the National Corrections Reporting Programs (NCRP) consisting of 10,907,333 rows and 18 variables. Our second sample dataset is the Chicago Crime (CC) consisting of 7,288,181 rows and 22 variables. (Refer to Appendix A&B for descriptions of the variables in both dataset). The following section details the steps adopted to prepare the data for preliminary and further analysis.

3.1 Data Cleaning

Before using the dataset for analysis, it is important to first clean it to correct or remove any erroneous and redundant data. We will also restructure some data for easier analysis.

3.1.1 Missing values

<u>CC dataset</u>

Although the dataset has a similar range of years from which it was mined (from 2001 to 2021), missing values was not a significant concern as dropping all rows with missing values constituted a total loss of about 10% of the dataset, which left us 6,604,676 rows.

<u>NCRP dataset</u>

As the dataset was mined from 1991 to 2014, it is no surprise to find a huge volume of missing data in the dataset. Most of the missing data are related to the year when the prisoner is going to be released *(Refer to Appendix C)*. For example, some columns such as PARELIG_YEAR are missing almost 80% of their values. Therefore, we cannot impute the values in these columns as more data is missing than available. As such, we have decided to drop all rows that have missing values. With that, we were left with 720189 rows to work with. Also, according to the data dictionary provided by NCRP, values with ‘9’ or ‘9999’ are an indication of missing values. Thus, we have also removed them from our dataset.

3.1.2 Logical Errors

CC dataset

There are no logical errors found in the dataset.

NCRP dataset

There are no logical errors found in the dataset as can be seen when all years of release are more than the year of admission.

3.1.3 Processing of dependent variables

CC dataset

There are two columns in the dataset that can identify the type of crime committed for each record of crime. There is a Primary Type column and a Description column. We have chosen to take Primary Type as our target variable as it specifically categorises the type of crime rather than giving it a description. Primary Type has a total of 35 unique categories of crime labels. It can be difficult for the model to accurately classify such a large number of categories. Furthermore, many types of crime categories are similar in nature. For example, Theft, Burglary and Robbery are three different categories labelled in the dataset. Therefore, we decided to simplify the prediction by reducing the number of categories. We chose to do this by grouping the types of crime based on how similar and related they are. For instance, cases of assault, battery and homicide were reclassified under the Primary Type ‘Assault’, while cases involving narcotics, gambling and alcohol were reclassified under ‘Addiction’. This enabled us to reduce the total number of classes for Primary Type from 35 to 7. The full reclassification of old to new classes can be found in Appendix C.

NCRP dataset

Since the data file contains one record for each separate term in prison, an individual person may have more than one record, but all will be assigned the same ABT_INMATE_ID value. Thus, we iterate through the dataset to find duplicate copies of ABT_INMATE_ID value which indicates that the felon has recommitted a crime. We will mark “0” as non-repeated offender and “1” as repeated offender for our dependent variable *recidivism (Refer to Appendix C).*

3.1.4 Redundant Variables

CC dataset

We chose to exclude several of the variables provided in the original dataset. The reasons for dropping them are listed below:

Variable	Reason
ID	ID and Case Number represent unique values to identify each instance of a crime that was recorded. They only serve to identify each case and have no meaningful value as a predictor.
Case Number	
Updated On	The date that each crime record was updated on is meaningless in the context of using it as an independent variable as it is simply administrative information.
X Coordinate, Y Coordinate, Latitude, Longitude, Location	Not useful unless interpreted on a map as they are just unique floating point values on their own. These variables represent exact geolocations and it is extremely unlikely that more than one instance of crime occurs at the exact same place. Hence, we have chosen to not make use of these location variables as they are too unique and specific to each crime record.

NCRP dataset

We removed ABT_INMATE_ID from the dataset as we realised it holds no value in our analysis as it only acts as an identifier for each criminal.

3.1.5 Scaling of continuous variables

CC dataset

Scaling of data is not required for this dataset.

NCRP dataset

The dataset consists of 5 columns that records a criminal’s Year of Admission, Year of Release, Year of Mandatory Prison Release, Year of Projected release, Year of Parole etc and have years extending all the way to more than 2100. As such, we treat them as continuous variables and perform a StandardScaler on these columns using the StandardScaler function *(Refer to Appendix C)* provided by the sklearn library. This normalizes the data to fall between a range of -1 and 1. This scaled data will then be used for our neural network model only.

3.1.6 Ensuring correct data type for each feature

CC dataset

Since most of our features are categorical variables (*refer to Appendix B*) but were originally of data types such as float and object (string), we have converted them to categorical using the .astype() function *(refer to Appendix C).*

NCRP dataset

Since most of our features are categorical variables (Refer to Appendix B), we have changed them to categorical using the .astype() function *(Refer to Appendix C).*

3.1.7 Checking for imbalance of data

CC dataset

The balancing of the data was achieved when we reclassified the primary types in the computation of the independent variable. The full classification can be found in Appendix C.

NCRP dataset

For our last data preparation step, we wrote a function to check for imbalance of data. Both cleaned datasets - with and without standard scaler - show that 54.45% of the prisoners did not recidivate while the remaining 45.55% did *(Refer to Appendix C)*. Thus, the dataset was quite balanced, and we can carry out our train-test split in preparation for our machine learning models.

4 Exploratory Analysis

After cleaning up the datasets, we conducted exploratory analysis to gain better understanding of our datasets.

4.1 Data exploration for CC dataset

We used Python to conduct data exploration on the CC dataset and have obtained some findings from our exploration. Most of the data exploration plots can be found in Appendix D, but one of our findings is as follows:

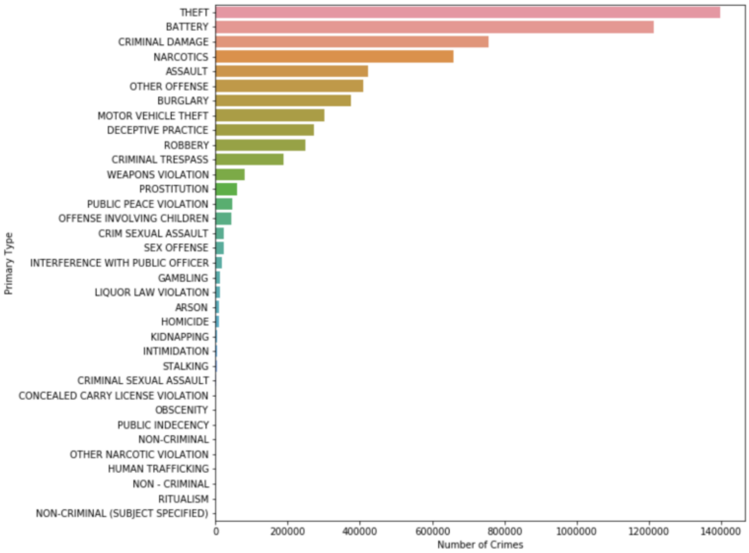


Figure 1. Barplot ranking the Primary Types of crime based on number of cases

In Figure 1 above, we can observe that the bottom half of the Primary Types of crime is much less than that of Primary Types such as Theft and Battery. This imbalance in the data prompted us to conduct a reclassification of the Primary Types in order to obtain a more balanced dataset with less dependent classes, as shown in Figure 2 below:

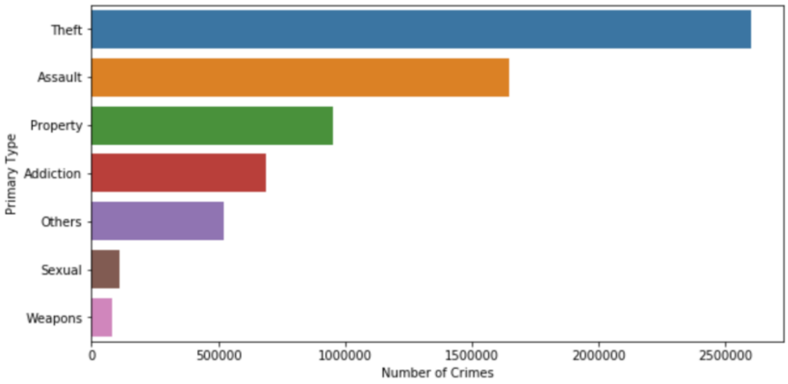


Figure 2. Barplot after reclassification

4.2 Data exploration for NCRP dataset

We used Python to conduct data exploration on the CC dataset and have obtained some findings from our exploration. Most of the data exploration plots can be found in Appendix D, but here are some of the findings we found interesting:

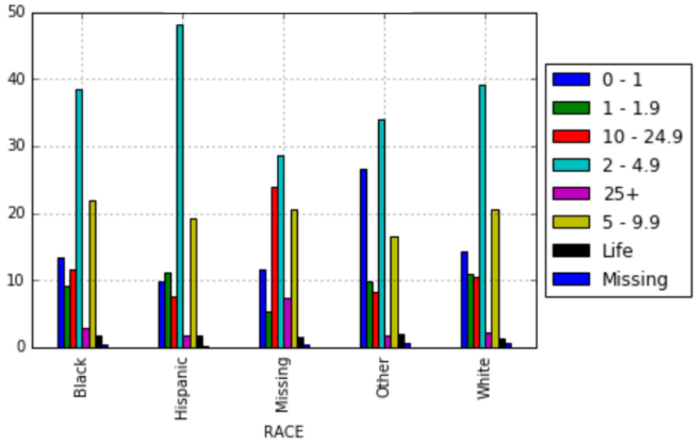


Figure 3. Bar graph showing the sentence length by race

With the prevailing racism in the United States, we have decided to explore the dataset to see if there are any racial biases when sentencing. However, from Figure 3, it was surprising to see that the sentences are fairly equal among races, taking into account the difference in number of offenders of races.

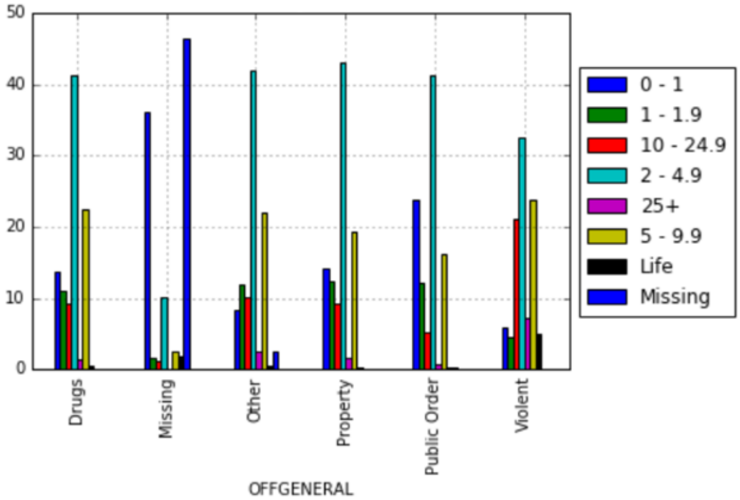


Figure 4. Bar graph showing the sentence length by offence

As we can see from Figure 4, *2-4.9 years* is a common sentence length for all types of offences. We can also notice that the violent cases tend to have higher sentences, especially life sentences.

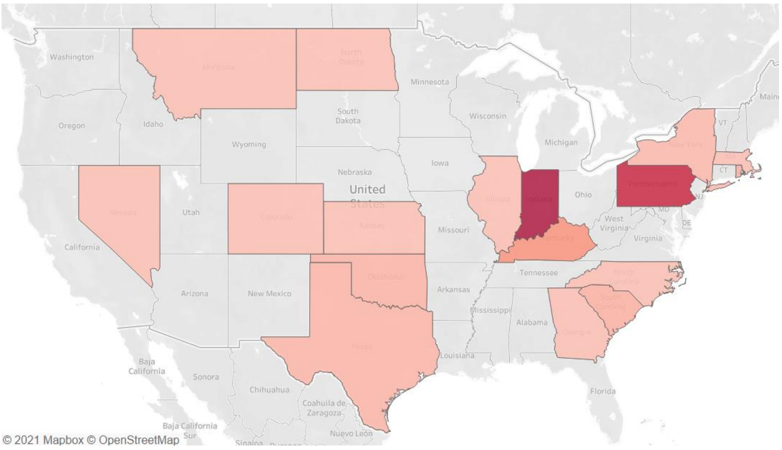


Figure 5. Map showing the state with the highest recidivism rate

Figure 5 shows that Indiana and Pennsylvania have especially high recidivism rates as compared to other states in the United States. This may be useful in reducing recidivism rates especially if the relevant authorities are able to look into the reason for higher recidivism rates in these states.

5 Proposed Solution

We will now provide an overview of our proposed **multi-pronged approach** to predict the following:

- Primary Type of crime that will happen in Chicago in the future
- Likelihood of a past offender committing another crime in the future

5.1 Predicting Primary Type of Crime

In this section, we will be analyzing historical records to predict the **primary type of crime that will happen**.

5.1.1 Overall Approach

We will be using two analytical models - namely the Classification and Regression Tree (CART) and Logistic Regression model to predict the type of crime committed (Primary Type).

5.1.2 Classification and Regression Tree (CART)

The CART model will be based on a classification tree as the variable to be predicted, Primary Type, is a variable with 7 categories. The DecisionTreeClassifier model from the Python Scikit-Learn library was used to construct the model.

The first classification tree was created using default parameter values for DecisionTreeClassifier: (*criterion='gini'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features=None*, *ccp_alpha=0.0)*

Using this model to predict Primary Type on both the train-set and test-set generated the following accuracy results:

Train-set Accuracy: 100.00%
Test-set Accuracy: 99.9975%

We use this model as the baseline for comparison with other models built using different parameter values. As the train-set accuracy for the baseline model is 100%, the model seems to be overfitted with training data despite having a high test-set accuracy. To reduce overfitting, hyperparameter tuning was done to find a more optimal model. We decided to use the cost complexity parameter to do post pruning on the classification tree using cost complexity pruning. The pruning technique built into DecisionTreeClassifier uses the hyperparameter ccp_alpha. The full hyperparameter tuning process done can be found in Appendix E.

The greater the value of ccp_alpha parameter, the number of nodes pruned from the tree increases, reducing the complexity of the model and how much it learns from the training data. The following plot shows how the train-set and test-set accuracies of the tree vary when ccp_alpha increases.

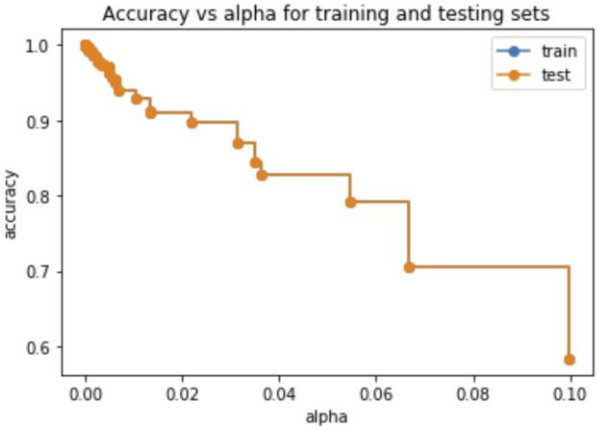


Figure 6. Accuracy vs alpha for training and testing sets for CART model

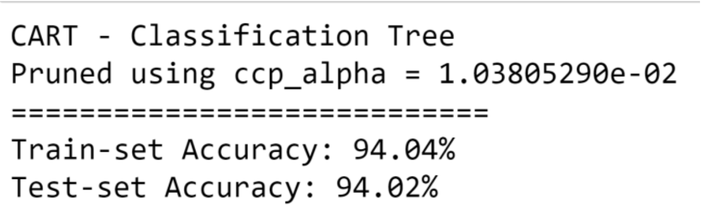
As seen from Figure 6, the train-set accuracy is almost identical to the test-set accuracy. The lower the value of ccp_alpha, the higher the accuracy of the model. While we would usually choose the optimal ccp_alpha value that gives the best possible validation accuracy, in this case, a classification tree built with ccp_alpha value of about 0.01 manages to achieve about 94% accuracy while being far less complex than if the most optimal ccp_alpha value was used.

Ultimately, we chose to build the final model using ccp_alpha = 1.03805290e-02 as it strikes an optimal balance between model accuracy and complexity. There were 549 nodes in the baseline model versus only 25 nodes in the final pruned model. Occam's Razor theory also supports this decision by stating that for two similar outcomes, the simpler method is more desirable (Brownlee, 2020). In this case, the 6% difference in validation accuracy is compensated by the much simpler model complexity between the baseline and pruned models, hence the pruned model was chosen as our final CART model.

The full final classification tree model can be found in Appendix E.

5.1.2.1 Classification and Regression Tree (CART) Results

Using a 70%-30% splitting ratio for train-set and test-set respectively and a fixed random_state parameter value to generate the exact sets of training and validation datasets each time, the final CART model produces the following training and validation results:



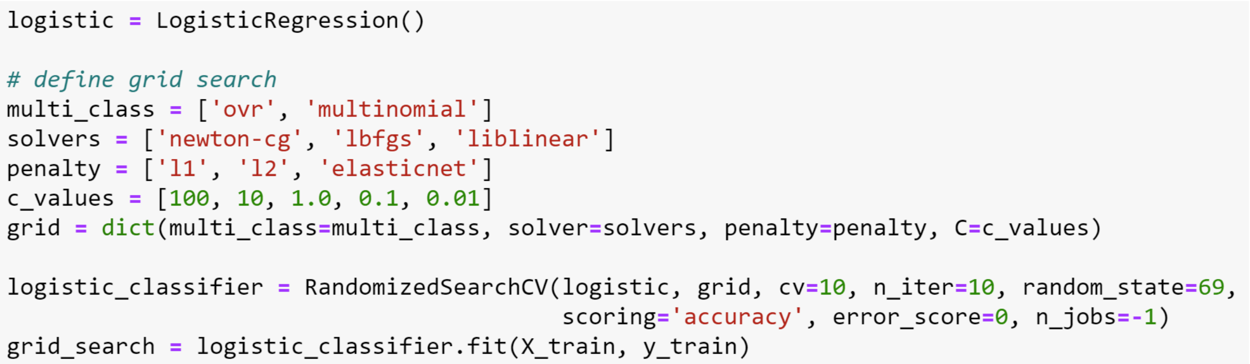
```
CART - Classification Tree
Pruned using ccp_alpha = 1.03805290e-02
============================
Train-set Accuracy: 94.04%
Test-set Accuracy: 94.02%
```

Figure 7. CART model accuracy

5.1.3 Logistic Regression Model

A multi-class Logistic Regression model can be tuned from a selection of hyperparameters which allows the model to be best customised for this task and dataset. For instance, the model's *solver* parameter comprises hyperparameters such as *newton-cg, lbfgs* and *liblinear*. However, it is not always clear which hyperparameters to select in order to achieve the best model performance.

As a result, we have utilised hyperparameter tuning using scikit-learn's *RandomizedSearchCV* library. This technique evaluates models for a given hyperparameter vector using a 10-fold cross-validation, as seen below in Figure 8.

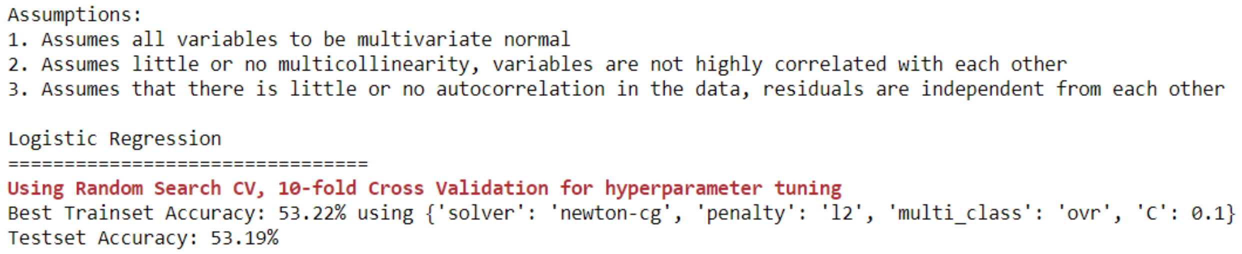


```python
logistic = LogisticRegression()

# define grid search
multi_class = ['ovr', 'multinomial']
solvers = ['newton-cg', 'lbfgs', 'liblinear']
penalty = ['l1', 'l2', 'elasticnet']
c_values = [100, 10, 1.0, 0.1, 0.01]
grid = dict(multi_class=multi_class, solver=solvers, penalty=penalty, C=c_values)

logistic_classifier = RandomizedSearchCV(logistic, grid, cv=10, n_iter=10, random_state=69,
                                         scoring='accuracy', error_score=0, n_jobs=-1)
grid_search = logistic_classifier.fit(X_train, y_train)
```

Figure 8. Hyperparameter Tuning using RandomizedSearchCV

5.1.3.1 Logistic Regression Model Results

Using the same training and testing dataset as from the CART model, a correlation heatmap was created to determine if there is multicollinearity within the predictive variables and the variable “Description” was dropped due to a high correlation coefficient with “IUCR”. The correlation heatmap can be found in Appendix D. The logistic regression model was then trained with the optimal hyperparameters and the model accuracy is shown in Figure 9 below:



```
Assumptions:
1. Assumes all variables to be multivariate normal
2. Assumes little or no multicollinearity, variables are not highly correlated with each other
3. Assumes that there is little or no autocorrelation in the data, residuals are independent from each other

Logistic Regression
================================
Using Random Search CV, 10-fold Cross Validation for hyperparameter tuning
Best Trainset Accuracy: 53.22% using {'solver': 'newton-cg', 'penalty': 'l2', 'multi_class': 'ovr', 'C': 0.1}
Testset Accuracy: 53.19%
```

Figure 9. Logistic Regression Model Assumptions and Results

The 7x7 confusion matrix can be found in Appendix E.

5.1.4 Analysis, Evaluation and Suggestions

From the final CART model, the feature importance shown in Appendix E that only 4 variables were used in the splitting decisions, with their importances in the following order: FBI Code, Description, IUCR, Arrest. This is an interesting observation as it shows that location and timing related information is not as significant as one might assume when predicting the type of crime occurring.

From Figure 9 above, it can be concluded that the CART model is the superior model to logistic regression based on accuracy in this context. CART offers about a 45% increase in accuracy over logistic regression. Therefore, CART is a more suitable model than Logistic Regression for predicting types of crime based on the data provided from the CC dataset. Other than having a higher accuracy, decision trees are also simpler and more intuitive to interpret as it follows a logical sequence based on splitting conditions at each non-terminal node.

One reason for the low accuracy of the Logistic Regression model may be due to our goal of predicting the Primary Type of crime, which is not a binary dependent variable. Logistic Regression is typically used for predicting outcomes with 2 categories by searching for a single linear decision boundary in the feature space. With multinomial Logistic Regression, the approach is to transform the problem into multiple binary classification problems (i.e. *one-vs-rest* or *one-vs-one*) but the data points may not be easily separated by a single hyperplane. Linearly separable data is rarely found in real-world scenarios. On the other hand, a classification tree is able to classify using non-linear decision boundaries and is a more flexible model.

One benefit of the Logistic Regression model lies in its simplicity and transparency. efficient to train and the variable coefficients can show how much the target variable changes from a change in any predictor variable. Therefore, a suggestion would be to use the Logistic Regression model as a benchmark for the CART model to compare and explore the extent to which types of crime changes due to changes in independent variables.

5.1.5 Limitations

For CART, the model relied heavily on the FBI Code variable as it was used for many of the splitting decision rules. However, information such as FBI Code, IUCR and Arrest made are usually not available before or at the time of the crime being initially reported. Hence, potential important information that the CART model uses may be unavailable in situations where law enforcement are attempting to predict future crimes or respond to real-time crimes. Since statistical imputation is not suitable for such attributes in this context, the data would have to be considered as missing. While one big advantage of CART models is their ability to handle missing data via surrogate splits where the next best splitting variable is used if it beats the majority rule, this may greatly reduce the model's accuracy results.

One possible improvement to the CC dataset to create better predictive models may be to include more predictive variables that are not directly related to the crime cases itself. Between the "Month", "Day" and "Hour" variables to the "Beat", "Ward" and "Community Area" variables, the dataset's predictive variables are largely centered within the time and place a crime had occurred previously. One example of a factor that can be considered to include in future datasets is the weather of the crime location at the time of reporting (Glorfeld, 2018). For example, climate change may lead to more murders (Ranson, 2013) and warmer ambient temperatures can lead to more sex offences (Xu et. al, 2021). With more attributes, the models generated can become more comprehensive and may unearth unexpected correlations and patterns between crime and other factors.

5.2 Predicting Likelihood of Criminal Recidivism

In this section, we will be predicting **whether a person is more or less likely to recommit a crime or commit a second felony.**

5.2.1 Overall Approach

We will be using Artificial Neural Networks (ANN) and Random Forest Classification to predict criminal recidivism. The ANN model that we are planning to use will be the widely used multi-layer perceptron, feedforward backpropagation method. We also adopt the Random Forest model in conjunction with the ANN to cross-check our accuracy. In addition, we will also be using the Random Forest model to determine the significant features that should be included in our NN model.

5.2.2 Building the Neural Network (NN)

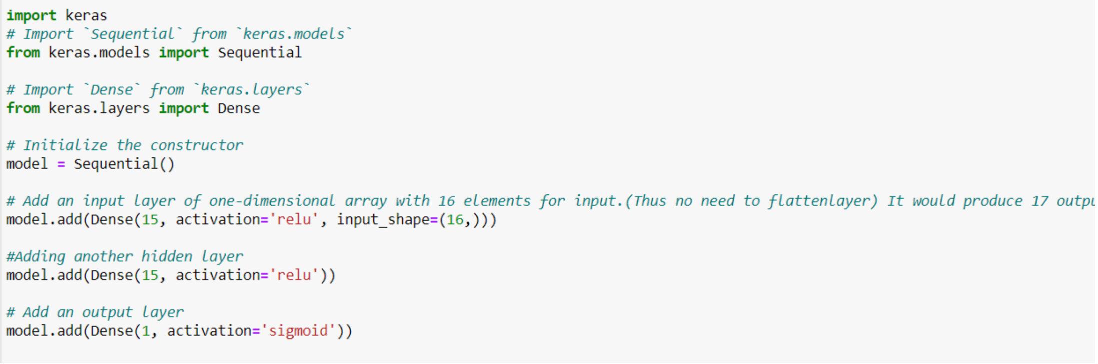
Before we can build our neural network, we have to first determine the two hyperparameters that control the architecture or topology of the model - the number of layers and the number of nodes in each hidden layer.

To begin with, we will be using the multilayer neural network. This is because a single layer neural network is limited to linearly separable functions whereby it is suitable for simple problems where the two classes in a classification problem can be neatly separated by a line. Our problem, however, might not be bounded linearly. Upon analysis, we have concluded that a multilayer neural network with two hidden layers would optimise our NN model. (Brownlee, J, 2019)

Next, we have to determine the number of nodes to be used in each layer. In the input layer, we used 16 nodes as we are using all 16 features in our dataset. For the output layer, we employ one node as we are trying to predict a binary classification problem. The book, "Introduction to Neural

Networks in Java”, recommends that the '*the optimal size of the hidden layer is usually between the size of the input and size of the output layers*'. Since the size of our input is 17 and the size of our output layer is 1, we have chosen 15 nodes for our hidden layer.

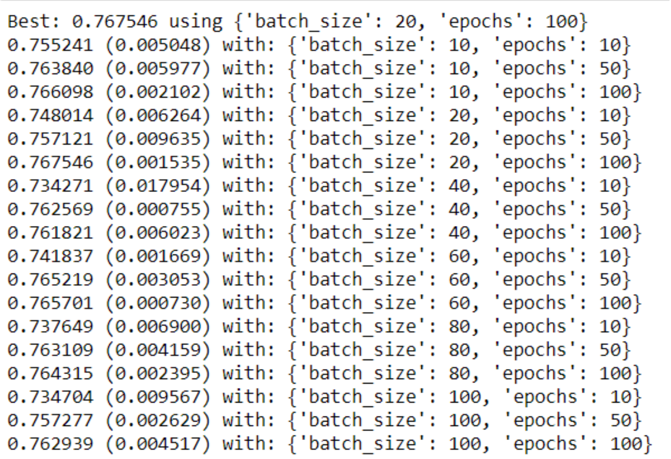
After determining the number of layers and nodes, we have to also determine the activation function that should be used in our neural network. For our input and hidden layers, we chose Rectified Linear Activation Function (RELU) as it prevents the emergence of the vanishing gradient problem which is common when using the Sigmoid or Tanh functions. For our output layer, we used the sigmoid function as we are trying to predict the probability of criminal recidivism which lies in the range of 0 to 1. (Sharma, S)



```
import keras
# Import `Sequential` from `keras.models`
from keras.models import Sequential

# Import `Dense` from `keras.layers`
from keras.layers import Dense

# Initialize the constructor
model = Sequential()

# Add an input layer of one-dimensional array with 16 elements for input.(Thus no need to flattenlayer) It would produce 17 outpu
model.add(Dense(15, activation='relu', input_shape=(16,)))

#Adding another hidden layer
model.add(Dense(15, activation='relu'))

# Add an output layer
model.add(Dense(1, activation='sigmoid'))
```

Figure 10. Building neural network model

Before we fit our neural network model to our training dataset, we will be using scikit-learn grid search capability. By defining the grid search parameters for batch size and epoch, we will iterate through and find the best possible combination of hyperparameters that gives the most accurate result. (Refer to Appendix E) In our case, the best combination would be a batch size of 20 and epoch of 100 as seen below.

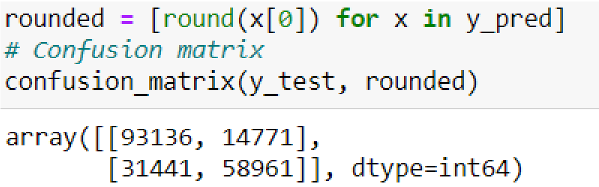


```
Best: 0.767546 using {'batch_size': 20, 'epochs': 100}
0.755241 (0.005048) with: {'batch_size': 10, 'epochs': 10}
0.763840 (0.005977) with: {'batch_size': 10, 'epochs': 50}
0.766098 (0.002102) with: {'batch_size': 10, 'epochs': 100}
0.748014 (0.006264) with: {'batch_size': 20, 'epochs': 10}
0.757121 (0.009635) with: {'batch_size': 20, 'epochs': 50}
0.767546 (0.001535) with: {'batch_size': 20, 'epochs': 100}
0.734271 (0.017954) with: {'batch_size': 40, 'epochs': 10}
0.762569 (0.000755) with: {'batch_size': 40, 'epochs': 50}
0.761821 (0.006023) with: {'batch_size': 40, 'epochs': 100}
0.741837 (0.001669) with: {'batch_size': 60, 'epochs': 10}
0.765219 (0.003053) with: {'batch_size': 60, 'epochs': 50}
0.765701 (0.000730) with: {'batch_size': 60, 'epochs': 100}
0.737649 (0.006900) with: {'batch_size': 80, 'epochs': 10}
0.763109 (0.004159) with: {'batch_size': 80, 'epochs': 50}
0.764315 (0.002395) with: {'batch_size': 80, 'epochs': 100}
0.734704 (0.009567) with: {'batch_size': 100, 'epochs': 10}
0.757277 (0.002629) with: {'batch_size': 100, 'epochs': 50}
0.762939 (0.004517) with: {'batch_size': 100, 'epochs': 100}
```

Figure 11. Determining epoch and batch size for neural network

5.2.2.1 Neural Network Model Results

Using a 70-30 train-test split, the model is trained with the optimal hyperparameters and the model results are shown below. *(Refer to Appendix E for the visual representation of the model)*



```
rounded = [round(x[0]) for x in y_pred]
# Confusion matrix
confusion_matrix(y_test, rounded)
```

```
array([[93136, 14771],
       [31441, 58961]], dtype=int64)
```

Figure 12. Confusion Matrix of the NN model

Accuracy Score	**0.77**
Precision Score	**0.80**
Recall Score	**0.65**
Specificity Score	**0.86**

Figure 13. Table of Results

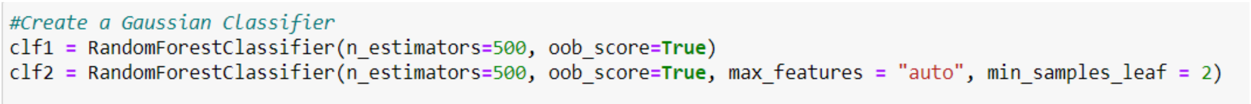
While the overall accuracy score is around 0.77, we will be prioritizing precision score rather than overall accuracy as we are trying to accurately identify criminals who are more prone to recommitting a crime so that more resources can be allocated to them. On that note, we have managed to achieve a precision score of 0.80.

5.2.3 Random Forest Model (RF Model)

To train the dataset with the Random Forest model, we will be using the RandomForestClassifier that is imported from the sklearn library. In addition to using accuracy scores to evaluate the performance of our Random Forest model, we will be validating the results of the model with the out-of-bag score - “oob_score”. Therefore, we specify “n_estimators = 500” which means that we will be using an ensemble of 500 trees and set “oob_score = True”.

A possible hyperparameter to explore is the min_samples_leaf hyperparameter. For our project, we trained one model with min_samples_leaf = 1 and the other with 2. When we increase the min_samples_leaf number, we can get a more stable average that we are calculating in each tree. This would result in each estimator to be less predictive but the estimators will also be less correlated so it can also help with overfitting problems, speed up training and make the model generalise better.

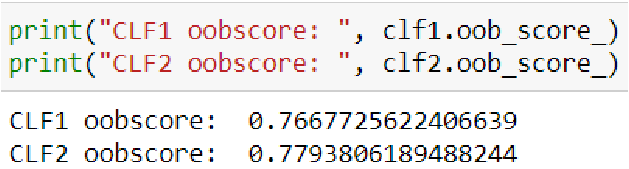
Another hyperparameter would be max_features, which helps to pick a different random subset of features at every decision point.Therefore, we can use max_features = “auto” to give us more variation which helps us to create more generalised trees that have less correlation.



```
#Create a Gaussian Classifier
clf1 = RandomForestClassifier(n_estimators=500, oob_score=True)
clf2 = RandomForestClassifier(n_estimators=500, oob_score=True, max_features = "auto", min_samples_leaf = 2)
```

Figure 15. Comparing two random forest models

High accuracy typically seen from the train set may be misleading as it might lead to an overfitting of the data. Therefore, we will look at the oob_score instead which calculates error on the training set, but only includes the trees in the calculation of a row’s error in which that particular row was not included in training that tree.



```
print("CLF1 oobscore: ", clf1.oob_score_)
print("CLF2 oobscore: ", clf2.oob_score_)
```

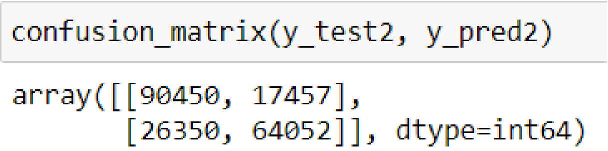
```
CLF1 oobscore:  0.7667725622406639
CLF2 oobscore:  0.7793806189488244
```

Figure 16. OOB scores of the two random forest models

As seen from Figure 16, we will be using min_samples_leaf = 2 and stating max_features = auto as they give us a better random forest model overall.

5.2.3.1 Random Forest Result

A train-test split with a ratio of 0.7-0.3 is performed before running the model. All variables from the dataset are passed into the RF model. The output are as follows:



```
confusion_matrix(y_test2, y_pred2)
```

```
array([[90450, 17457],
       [26350, 64052]], dtype=int64)
```

Figure 17. Confusion Matrix of the RF model

Accuracy Score	**0.78**
Precision Score	**0.79**
Recall Score	**0.71**
Specificity Score	**0.84**

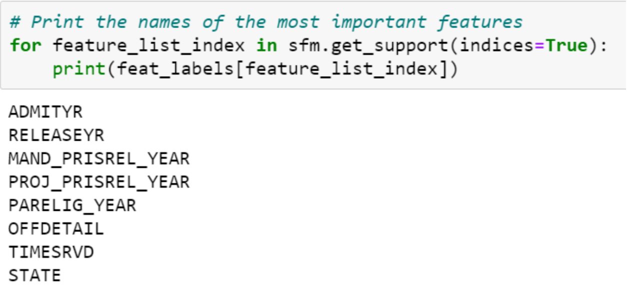
Figure 18. Table of Results

5.2.4 Determining Significant Features

In our project, we hope to be able to fine tune and reduce the number of features that the NN is currently employing through an integrated process of combining the feature selection process from other models with the NN model.

Literature shows that among the machine learning techniques, the Random Forest model has been an excellent tool to learn feature representations given their robust classification power and easily interpretable learning mechanism (Kong & Yu, 2018). As Random Forest is a supervised learning algorithm that builds an ensemble of decision trees that are usually trained with the "bagging" method, it can help to achieve sparse learning with less parameters in the NN model. The importance of features in each base learner can be easily obtained in the Random Forest model. As such, we have decided to use the Random Forest Model to extract more important features that we can integrate into our RF and NN model.

By using the "SelectFromModel" object from sklearn that can automatically select important features, we obtain the following results for the Random Forest model as follows:



```
# Print the names of the most important features
for feature_list_index in sfm.get_support(indices=True):
    print(feat_labels[feature_list_index])
```

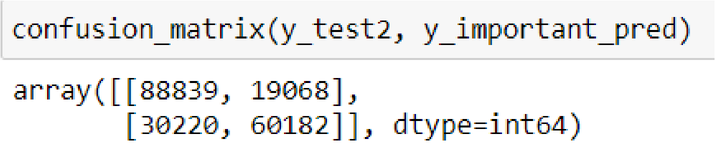
```
ADMITYR
RELEASEYR
MAND_PRISREL_YEAR
PROJ_PRISREL_YEAR
PARELIG_YEAR
OFFDETAIL
TIMESRVD
STATE
```

Figure 19. Feature selection result from Random Forest model

Next, we fit these selected important features into our models for analysis.

RF model:

We used SelectFromModel's in-built function .transform() to create a new dataset containing only the most important features. Below are the results.



```
confusion_matrix(y_test2, y_important_pred)
```

```
array([[88839, 19068],
       [30220, 60182]], dtype=int64)
```

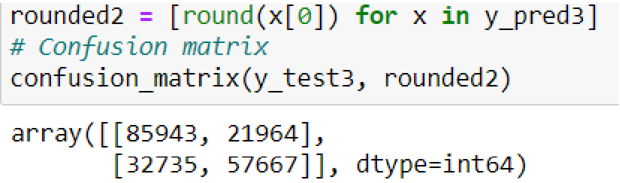
Figure 20. Confusion Matrix of the RF model (revised)

Accuracy Score	**0.75**
Precision Score	**0.76**
Recall Score	**0.67**
Specificity Score	**0.82**

Figure 21. Table of Results

NN model:

Since we will be using 8 features as our input, we will change our number of nodes accordingly. *(Refer to Appendix E for the visual representation of the model)*



```
rounded2 = [round(x[0]) for x in y_pred3]
# Confusion matrix
confusion_matrix(y_test3, rounded2)
```

```
array([[85943, 21964],
       [32735, 57667]], dtype=int64)
```

Figure 22. Confusion matrix of the new NN model

Accuracy Score	**0.72**
Precision Score	**0.72**
Recall Score	**0.64**
Specificity Score	**0.80**

Figure 23. Table of Results

As seen above, every score for the NN model decreased after performing feature selection.

5.2.5 Analysis, Evaluation and Suggestions

The summarised test set results of the four neural network models are tabulated as shown below:

Results	**NN model**	**NN model (revised)**	**RF model**	**RF model (revised)**
Accuracy Score	0.77	0.72	0.77	0.75
Precision Score	0.80	0.72	0.76	0.76
Recall Score	0.65	0.64	0.71	0.67
Specificity Score	0.86	0.80	0.81	0.82

Based on the comparison in the table above, we can see that the RF model has the highest overall accuracy as compared to the rest. While feature selection does indeed improve in terms of the model complexity, it does not necessarily guarantee improvement in classification accuracy. In fact, there is a tradeoff between accuracy and execution time when it comes to feature selection. In our case, that is especially so since both models experienced a drop in accuracy after being subjected to feature selection. For the case of neural networks, feature engineering is already being executed in the back end of the neural network model to improve the model's accuracy. By conducting a separate feature selection, it will further decrease the feature space and lead to a less accurate predictive model due to the loss of important patterns to be recognised. Thus, reducing the number of features might harbour a counter-intuitive effect instead.

Previously, we have already mentioned that we will be prioritizing precision score (TP/TP+FP) rather than overall accuracy as we are trying to accurately identify criminals who are more prone to recommitting a crime so that more resources can be allocated to them. By looking across all 4 models, we can determine that the NN model before feature selection is the best.

Last but not the least, some might purport that since the NN model requires us to mine 16 different features, it might not be practical and feasible. However, that is not the case in our analysis since all of the data can be easily obtained as a criminal is obligated to surrender all his/her personal documentations when charged.

In conclusion, the NN model should be used as the prediction model. From our analysis, the good performance of the NN model does not seem to be due to overfitting, but a genuinely good model with both low bias and low variance.

5.2.6 Limitations

Given that the neural network model wins in comparison compared to the rest, there are some potential limitations in the model we built to predict criminal recidivism. Admittedly, while neural networks boast a higher accuracy in predictions, it is harder to interpret due to its black box nature. This is because neural networks are systems that are built to arrive at complex answers and to do so, they generate their own questions and rules. (Robbins, B, 2017) This lack of transparency might be an issue when it comes to criminal recidivism. Will a criminal be satisfied by knowing that it has been judged by an artificial system that it will recommit a crime again in the future? Will the criminal be dealt with fairly and justly if the decision is solely based on this model?

6 Further Actions

6.1 Implementing Multi-Task Classification and Multi-Label Classification

For the CC problem of predicting Primary Type, our classification models may consider the implementation of multi-task classification and/or multi-label classification approaches. Multi-Task Classification (also known as Multiclass-multioutput classification) is a classification task which labels each sample with a set of non-binary properties.(Scikit-learn, 2011) For example, a multi-task classification may be to predict the Primary Type **and** whether an arrest had been made simultaneously.

Multi-label classification is a predictive modeling task that involves predicting zero or more mutually non-exclusive class labels (Brownlee, 2020). For our current prediction of Primary Type, our multi-class classification predicts classes that are mutually exclusive (Nooney, 2019). However, we know that in reality, many crimes that happen involves multiple charges such as murder and rape. In such a case, will the crime be classified as murder or sexual assault? Therefore, implementing multi-label classification would help to solve this problem as it will be able to produce a prediction with both labels. We can implement multi-label classification by importing LabelPowerset from skmultilearn.problem_transform or MLkNN from skmultilearn.adapt.

6.2 Extendability

One area that is worth exploring is the prediction of crime location hotspots where police can employ a spatiotemporal crime prediction technique based on machine learning coupled with 2-Dimensional Hotspot analysis. (Hajela, G., Chawla, M., & Rasool, A, 2020, April 16). This is important when it comes to predictive policing as police are able to prevent crime before it even happens.

6.3 Other Considerations for Jurisdiction - Literature Review and Expert Opinion

No predictive model is perfectly accurate - the decision to commit crime is innately complex and the intention of crime can never be fully known by law enforcement. Therefore, in this project where we used a predictive model to recognise associations and patterns from historical offenders data that can help policing agencies to better analyse each case, there exists a need to combine our model results with literature review and expert opinion. In our prediction of whether a criminal would recommit a crime, a judge’s idea may be to lengthen the sentence length to deter the offender’s intention of recidivism, however, this decision should be backed up with expert opinion. For example, forensic psychologists can be engaged during jurisdiction to help in understanding why certain behaviours occur, and also in helping to minimize and prevent such behaviours. The forensic psychologist can suggest other ways of punishment that are suited to their evaluation of the offender. Furthermore, in crimes that consist of a specialised discipline or technical nature, experts of that field should be consulted on their opinion of the intention of the offender and the severity of the crime. Therefore, expert opinion should be consulted to make more educated decisions in charging a criminal.

7 Singapore Context

7.1 Criminal Situation in Singapore

In Singapore, the total number of reported crimes went up by 11.6% to 18,121 cases in the first half of 2020, compared to 16,240 cases reported in the same period in 2019. This increase was primarily due to the rise in scam cases, with online scams seeing a significant increase. On the other hand, physical crime dropped by nearly 2,000 cases across three broad crime classes - crimes against persons such as serious hurt, outrage of modesty and rioting, housebreaking and related crimes, as well as theft (CNA, 2020).

As for recidivism in Singapore, the overall recidivism rate for the 2018 release cohort was 22.1%, which is a decrease from 24% in 2017 and 23.7% in 2016. According to the Singapore Prison Service, the recidivism rate has dipped to an all-time low and more inmates are serving part of their jail term in the community (Mahmud, 2021).

7.2 Data Exploration for Singapore Criminal Situation

Upon data exploration on the limited data available, we can see that cheating related offences are on the rise as compared to most of the other physical crimes.

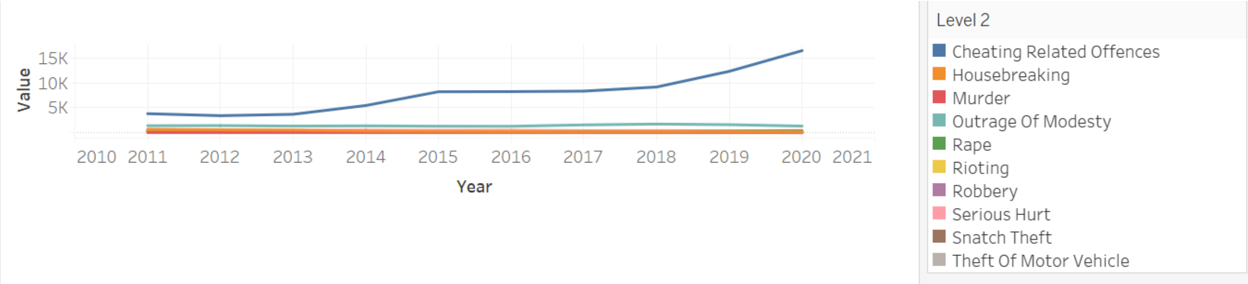


Figure 24. Year vs. type of offences in Singapore

From literature review and data exploration on limited data, we can conclude that cheating offences should be an area of concern for criminal offences in Singapore. As for factors that contribute most to such offences, more data are required for further analysis.

7.3 Relevance of Project to Singapore Context

The primary difference between our project and the situation in Singapore would be the type of crimes that is on the rise, with virtual scams being more of a threat locally. Additionally, one of our main predictions for the project - recidivism - is also not a big concern in Singapore. Therefore, we can extend our project to better cater to the local situation.

As compared to rising physical crimes in Chicago which requires policing agencies to make better decisions when dispatching assistance, Singapore is concerned with more virtual crimes such as online scams and radicalisation. As technology advances, online scammers have also been innovating new techniques to make their scams more realistic. Hence, to help the local policing agencies to better identify these scams, we can make use of appropriate datasets which contain information of the online scams that happened and train a predictive model to predict future online scam cases. As the model is trained more often with improved datasets, more relevant variables leading to higher accuracy of the predictive model can help to identify newer types of online scams more accurately.

Moreover, we can also use text mining to extract important concepts or terms from police statements and reports on these online scams and check if there are existing correlations between the found items and other crime characteristics (Ananyan, Sergei, 2004). Moving forward, we can perform pattern analysis where all the extracted items are used for tagging individual reports, allowing for further usage of these terms as new structured attributes, potentially identifying new scamming methods.

As recidivism rate for Singapore is low and measures such as a calibrated rehabilitation approach has been taken to minimise re-offending, we can modify our predictive model for criminal recidivism to cater to the local context, such as predicting if an offender would require jail term as punishment and rehabilitation. This prediction can not only help to better utilise the public service resources, but also gives the individual a second chance at life without the stigma of a prison record.

However, due to current data limitations, the above suggestions will only be possible if more detailed data are available for analysis.

8 Conclusion

By running our chosen datasets for both crime and recidivism statistics through multiple potential data analytics models, we conclude that at a rudimentary level, CART and Neural Networks can provide an accurate way to predict crime in a city.

By predicting and pinpointing recidivists helps not only in efficient policing but opens up new arenas for more holistic, targeted law enforcement in the future - finding correlations between certain characteristics, like income and education level allows for insights towards the notion that the some crimes may be systemic (Bunge, M., 2006). This may lead to governance that is focused more easing burdens that lead to crime rather than a judiciary focused on punitive measures. Governments and agencies may use data on recidivism to introduce changes to communities from the ground up, so that the motivation to commit crimes and re-offend upon release can be eradicated from the root. This will not only build a safer and more equitable society, but one where the police and their resources are less strained.

9 Appendices

9.1 Appendix A: Data Dictionary for CC dataset

Variable	Description	Data Type
ID	Case ID (unique to each case)	String
Case Number	Case number (unique to each case)	String
Date of occurence	Date when incident occurred	String
Block	The partially redacted address where the incident occurred	String
IUCR (Illinois Uniform Crime Reporting Program)	Directly linked to the primary type and description. https://www.isp.state.il.us/docs/6-260.pdf	String
Primary Description	Primary description of the IUCR code	String
Secondary Description	Secondary description of the IUCR code, a subcategory of the primary description	String
Location Description	Description of the location where the incident occurred	String
Arrest	Indicates whether an arrest was made	Boolean
Domestic	Indicates whether the incident was domestic-related as defined by the Illinois Domestic Violence Act	Boolean
Beat	Indicates the beat where the incident occured. A beat is the smallest police geographic area - each beat has a dedicated police beat car	Integer
Ward	Ward (City Council district) where the incident occurred	Integer
Community Area	Community Area code of where the crime occurred. The city of Chicago is divided	Integer

	into 77 community areas	
FBI CD	Indicates the crime classification as outlined in the FBI's National Incident-Based Reporting System (NBIRS)	String
X-coordinate	X-coordinate of where the crime occurred	Float
Y-coordinate	Y-coordinate of where the crime occurred	Float
Updated On	Date when record was last updated	String
Latitude	Latitude of where the crime occurred	Float
Longitude	Longitude of where the crime occurred	Float
Location	Latitude and longitude of where the crime occurred	String
Community Area Name	The name of the community area corresponding to its community area code	String

9.2 Appendix B: Data Dictionary for NCRP dataset

Variables	**Description**
ABT_INMATE_ID	Inmate identification ID
SEX	Sex of inmate ● 1 = Male ● 2 = Female
ADMTYPE	Type of prison admission ● 1 = New court commitment ● 2 = Parole return/revocation ● 3 = Other admission (including unsentenced, transfer, AWOL/escapee return) ● 9(M) = Missing
OFFGENERAL	5-level categorisation of most serious sentences offense ● 1 = Violent ● 2 = Property ● 3 = Drugs ● 4 = Public order ● 5 = Other/unspecified ● 9 (M) = Missing
EDUCATION	Highest level of education of inmate ● 1 = <HS diploma/GED ● 2 = HS diploma/GED ● 3 = Any college ● 99(M) = Ungraded/unknown
ADMITYR	Year inmate was admitted to prison ● 9999(M) = Missing
RELEASEYR	Year inmate was released from prison ● 9999(M) = Missing
MAND_PRISREL_YEAR	Year of mandatory prison release
PROG_PRISREL_YEAR	Year of projected prison release
PARELIG_YEAR	Year of parole eligibility
SENTLGTH	Maximum sentence length for inmate ● 0 = <1 year ● 1 = 1-1.9 years ● 2 = 2-4.9 years ● 3 = 5-9.9 years ● 4 = 10-24.9 years ● 5 = >=25 years

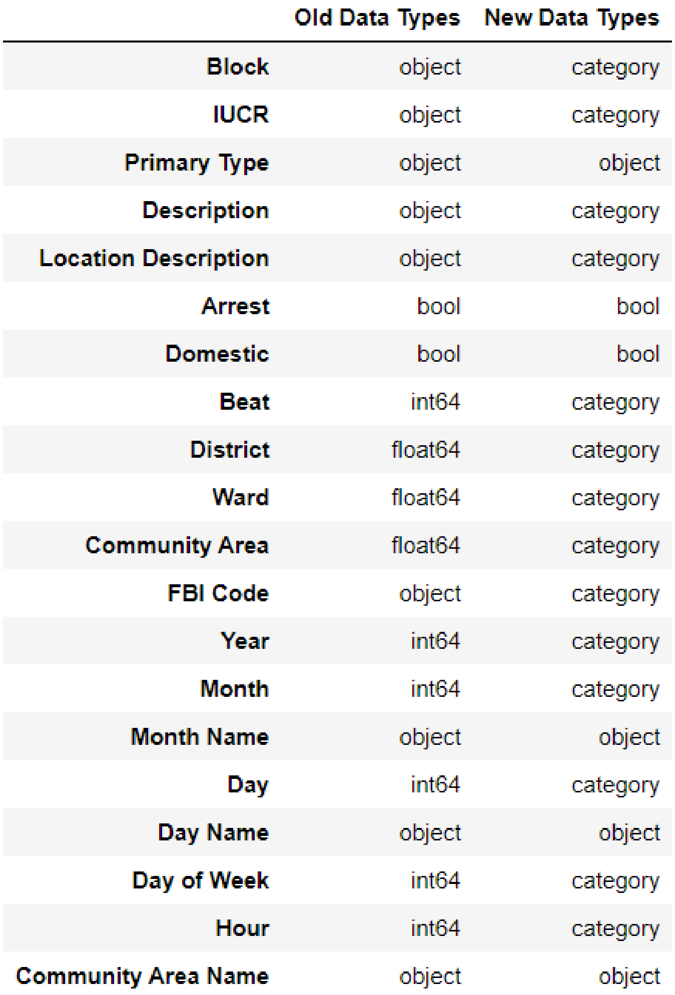
	● 6 = Life, LWOP, Life plus additional years, Death ● 9(M) = Missing
OFFDETAIL	Detailed categorisation of most serious sentenced offense ● 1 = Murder ● 2 = Negligent manslaughter ● 3 = Rape/Sexual Assault ● 4 = Robbery ● 5 = Aggravated or simple assault ● 6 = Other violent offenses ● 7 = Burglary ● 8 = Larceny ● 9 = Motor vehicle theft ● 10 = Fraud ● 11 = Other property offenses ● 12 = Drugs ● 13 = Public order ● 14 = Other/unspecified ● 99(M) = Missing
RACE	Race/hispanic ethnicity of inmate ● 1 = White ● 2 = Black ● 3 = Hispanic ● 4 = Other races ● 9(M) = Missing
AGEADMIT	Age at admission ● 1 = 18-24 years ● 2 = 25-34 years ● 3 = 35-44 years ● 4 = 45-54 years ● 5 = 55+ years ● 9(M) = Missing
AGERELEASE	Age at release ● 1 = 18-24 years ● 2 = 25-34 years ● 3 = 35-44 years ● 4 = 45-54 years ● 5 = 55+ years ● 9(M) = Missing
TIMESRVD	Time served by inmate ● 0 = <1 year ● 1 = 1-1.9 years ● 2 = 2-4.9 years ● 3 = 5-9.9 years ● 4 = >=10 years ● 9(M) = Missing

RELTYPE	Type of prison release ● 1 = Conditional release ● 2 = Unconditional release ● 3 = Other release ● 9(M) = Missing
STATE	State with custody inmate ● 1 = Alabama ● 2 = Alaska ● 4 = Arizona ● 5 = Arkansas ● 6 = California ● 8 = Colorado ● 9 = Connecticut ● 10 = Delaware ● 11 = District of Columbia ● 12 = Florida ● 13 = Georgia ● 15 = Hawaii ● 16 = Idaho ● 17 = Illinois ● 18 = Indiana ● 19 = Iowa ● 20 = Kansas ● 21 = Kentucky ● 22 = Louisiana ● 23 = Maine ● 24 = Maryland ● 25 = Massachusetts ● 26 = Michigan ● 27 = Minnesota ● 28 = Mississippi ● 29 = Missouri ● 30 = Montana ● 31 = Nebraska ● 32 = Nevada ● 33 = New Hampshire ● 34 = New Jersey ● 35 = New Mexico ● 36 = New York ● 37 = North Carolina ● 38 = North Dakota ● 39 = Ohio ● 40 = Oklahoma ● 41 = Oregon ● 42 = Pennsylvania ● 44 = Rhode Island ● 45 = South Carolina ● 46 = South Dakota ● 47 = Tennessee

	● 48 = Texas ● 49 = Utah ● 50 = Vermont ● 51 = Virginia ● 53 = Washington ● 54 = West Virginia ● 55 = Wisconsin ● 56 = Wyoming

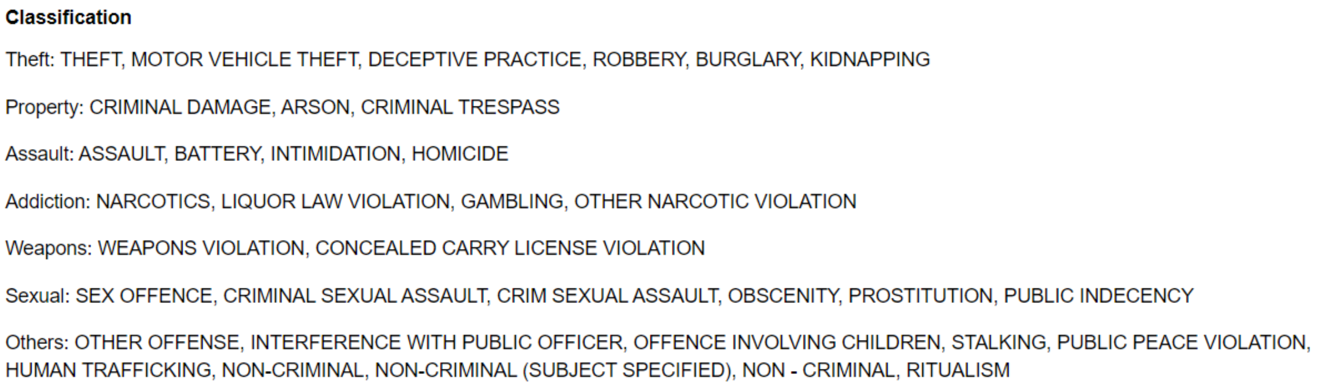
9.3 Appendix C: Data Preparation

CC Dataset



	Old Data Types	New Data Types
Block	object	category
IUCR	object	category
Primary Type	object	object
Description	object	category
Location Description	object	category
Arrest	bool	bool
Domestic	bool	bool
Beat	int64	category
District	float64	category
Ward	float64	category
Community Area	float64	category
FBI Code	object	category
Year	int64	category
Month	int64	category
Month Name	object	object
Day	int64	category
Day Name	object	object
Day of Week	int64	category
Hour	int64	category
Community Area Name	object	object

Figure 26. Conversion into correct data types



Classification

Theft: THEFT, MOTOR VEHICLE THEFT, DECEPTIVE PRACTICE, ROBBERY, BURGLARY, KIDNAPPING

Property: CRIMINAL DAMAGE, ARSON, CRIMINAL TRESPASS

Assault: ASSAULT, BATTERY, INTIMIDATION, HOMICIDE

Addiction: NARCOTICS, LIQUOR LAW VIOLATION, GAMBLING, OTHER NARCOTIC VIOLATION

Weapons: WEAPONS VIOLATION, CONCEALED CARRY LICENSE VIOLATION

Sexual: SEX OFFENCE, CRIMINAL SEXUAL ASSAULT, CRIM SEXUAL ASSAULT, OBSCENITY, PROSTITUTION, PUBLIC INDECENCY

Others: OTHER OFFENSE, INTERFERENCE WITH PUBLIC OFFICER, OFFENCE INVOLVING CHILDREN, STALKING, PUBLIC PEACE VIOLATION, HUMAN TRAFFICKING, NON-CRIMINAL, NON-CRIMINAL (SUBJECT SPECIFIED), NON - CRIMINAL, RITUALISM

Figure 27. Classification of Primary Types

NCRP Dataset

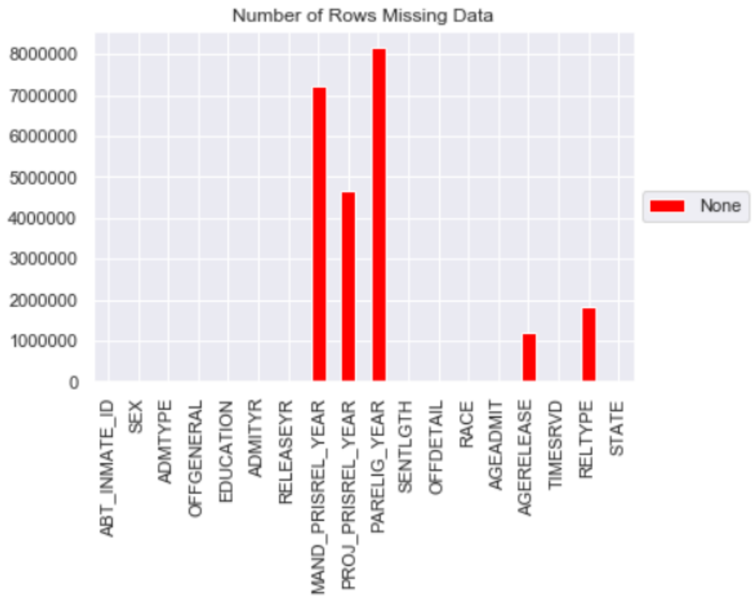
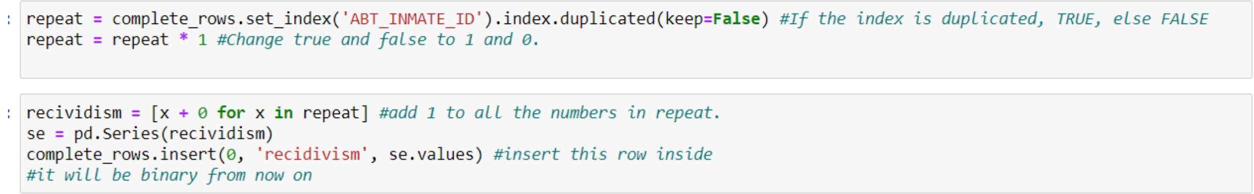


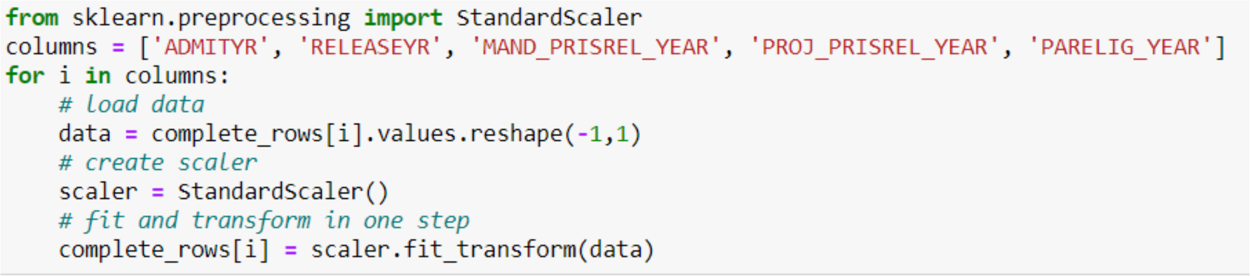
Figure 28. Graph Visualization of Missing Values



```
repeat = complete_rows.set_index('ABT_INMATE_ID').index.duplicated(keep=False) #If the index is duplicated, TRUE, else FALSE
repeat = repeat * 1 #Change true and false to 1 and 0.
```

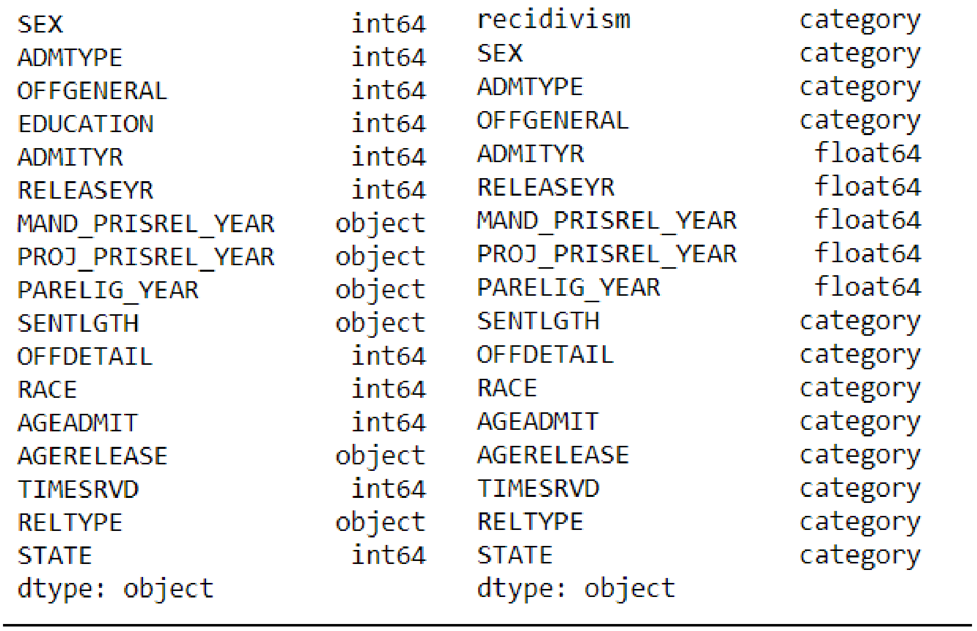
```
recividism = [x + 0 for x in repeat] #add 1 to all the numbers in repeat.
se = pd.Series(recividism)
complete_rows.insert(0, 'recidivism', se.values) #insert this row inside
#it will be binary from now on
```

Figure 29. Computation of Independent Variables



```
from sklearn.preprocessing import StandardScaler
columns = ['ADMITYR', 'RELEASEYR', 'MAND_PRISREL_YEAR', 'PROJ_PRISREL_YEAR', 'PARELIG_YEAR']
for i in columns:
    # load data
    data = complete_rows[i].values.reshape(-1,1)
    # create scaler
    scaler = StandardScaler()
    # fit and transform in one step
    complete_rows[i] = scaler.fit_transform(data)
```

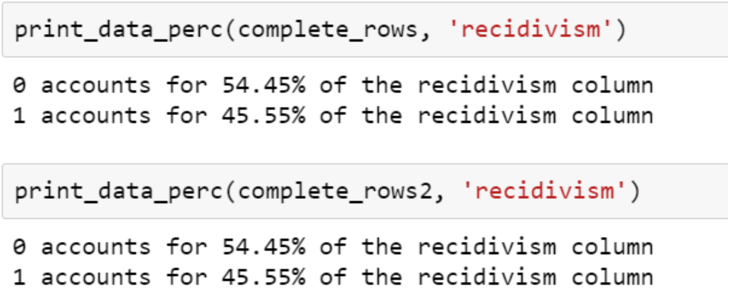
Figure 30. Standard Scaling of Continuous Variables



```
SEX                   int64
ADMTYPE               int64
OFFGENERAL            int64
EDUCATION             int64
ADMITYR               int64
RELEASEYR             int64
MAND_PRISREL_YEAR    object
PROJ_PRISREL_YEAR    object
PARELIG_YEAR         object
SENTLGTH             object
OFFDETAIL             int64
RACE                  int64
AGEADMIT              int64
AGERELEASE           object
TIMESRVD              int64
RELTYPE              object
STATE                 int64
dtype: object
```

```
recidivism           category
SEX                  category
ADMTYPE              category
OFFGENERAL           category
ADMITYR               float64
RELEASEYR             float64
MAND_PRISREL_YEAR     float64
PROJ_PRISREL_YEAR     float64
PARELIG_YEAR          float64
SENTLGTH             category
OFFDETAIL            category
RACE                 category
AGEADMIT             category
AGERELEASE           category
TIMESRVD             category
RELTYPE              category
STATE                category
dtype: object
```

Figure 31. Before and After Data Types of Our Dataset



```
print_data_perc(complete_rows, 'recidivism')
```

```
0 accounts for 54.45% of the recidivism column
1 accounts for 45.55% of the recidivism column
```

```
print_data_perc(complete_rows2, 'recidivism')
```

```
0 accounts for 54.45% of the recidivism column
1 accounts for 45.55% of the recidivism column
```

Figure 32. Checking for imbalance in dataset

9.4 Appendix D: Data Exploration and Visualisation

CC dataset

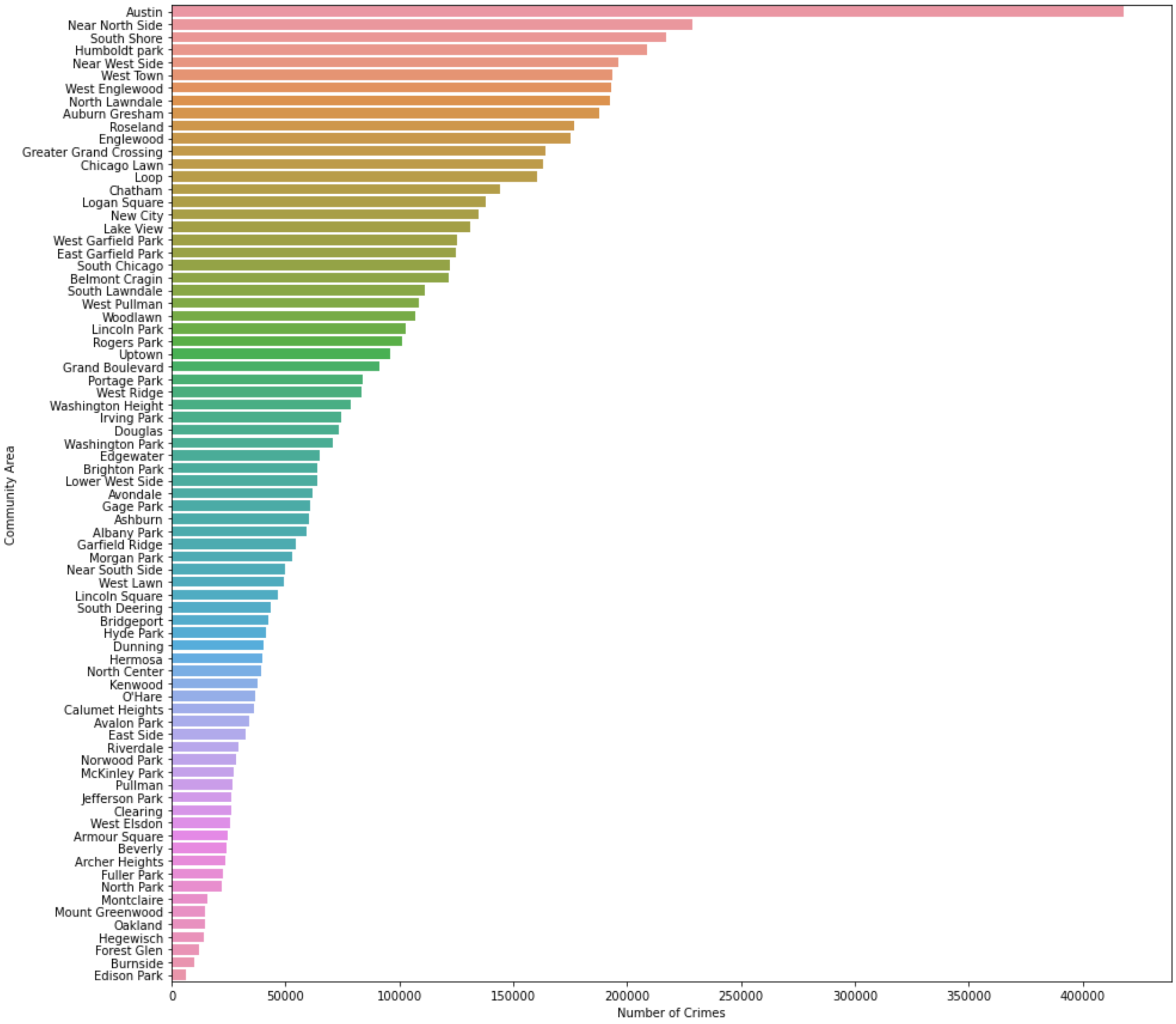


Figure 33. Ranking each Community Area based on the number of crimes

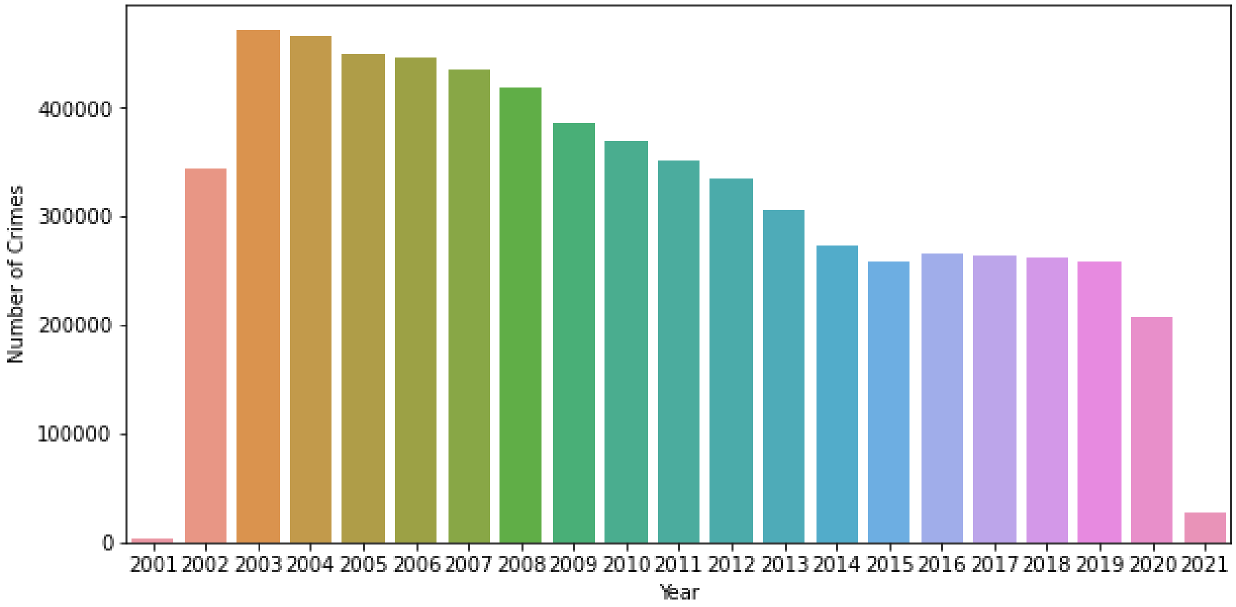


Figure 34. Trend of the number of crimes over the years

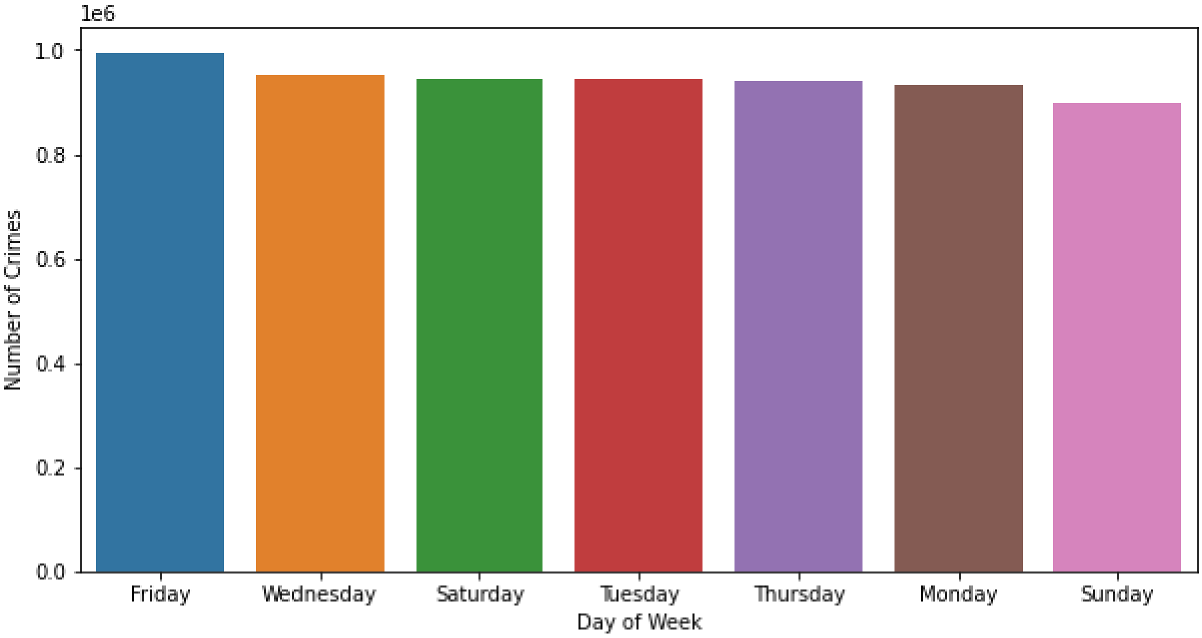


Figure 35. Determining which day has the most number of crimes

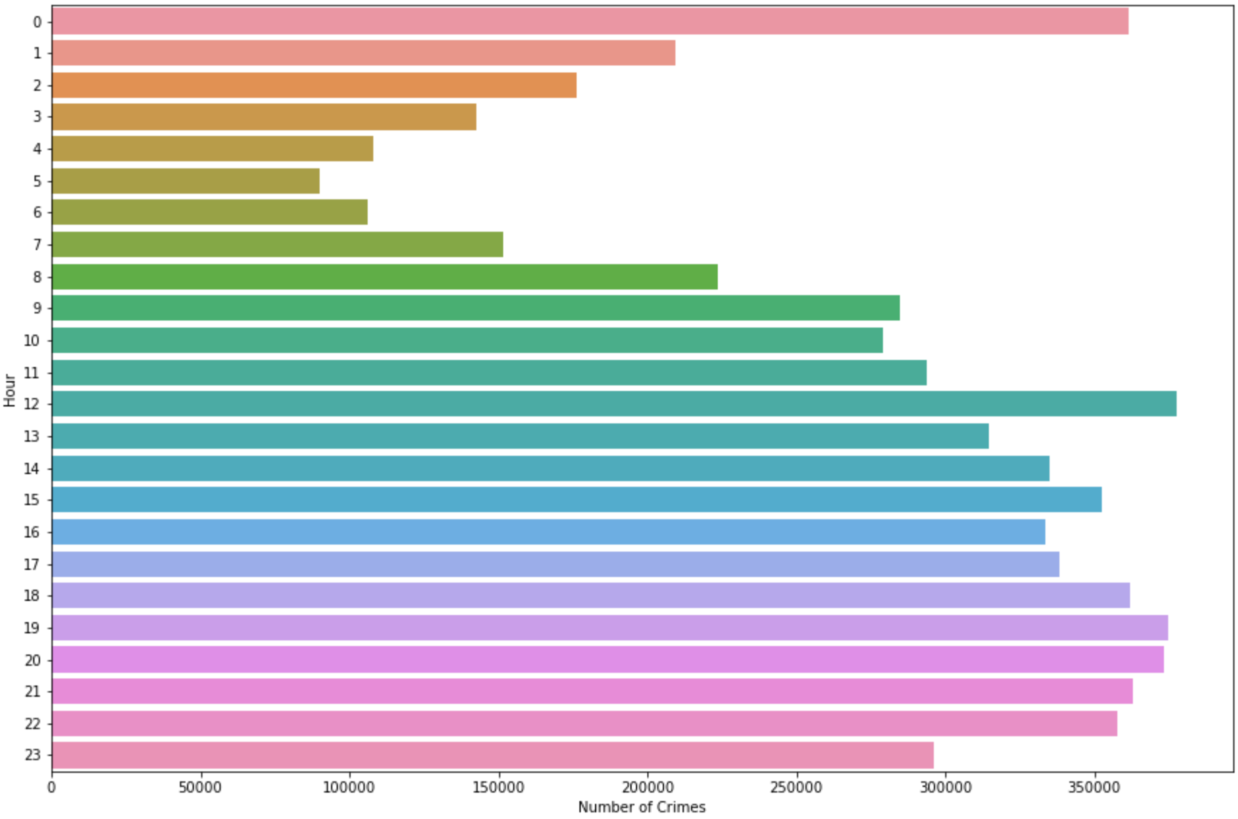


Figure 36. Determining which hour has the most number of crimes

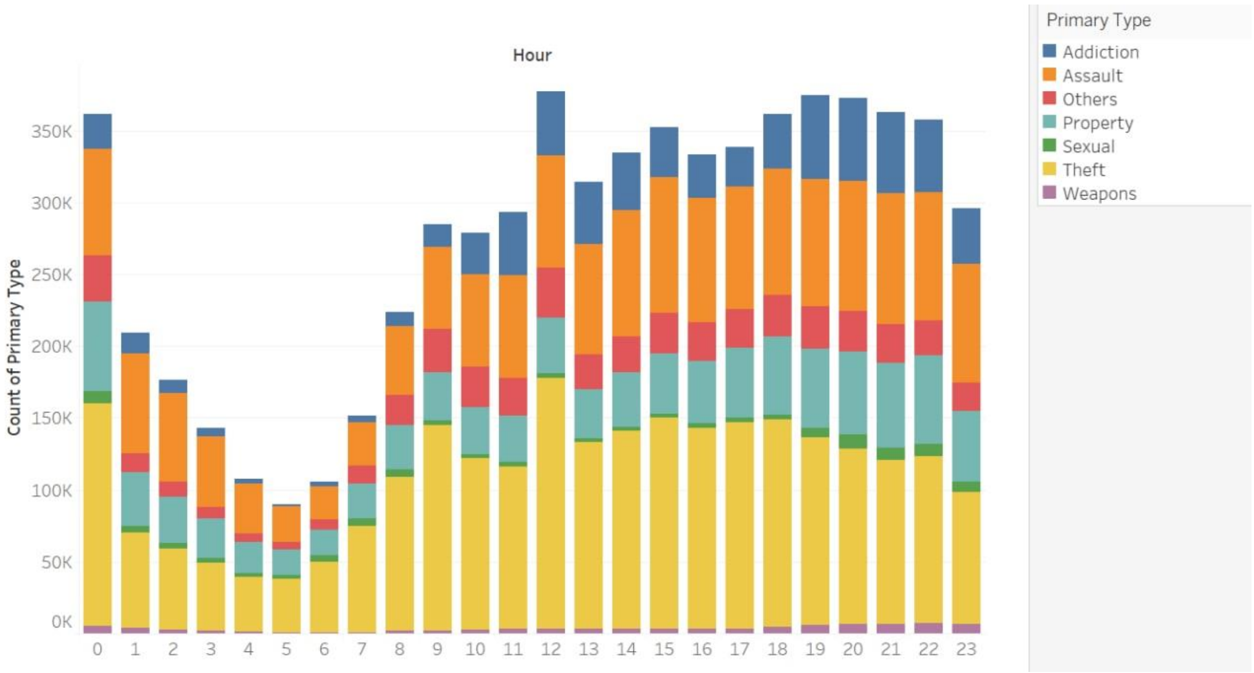
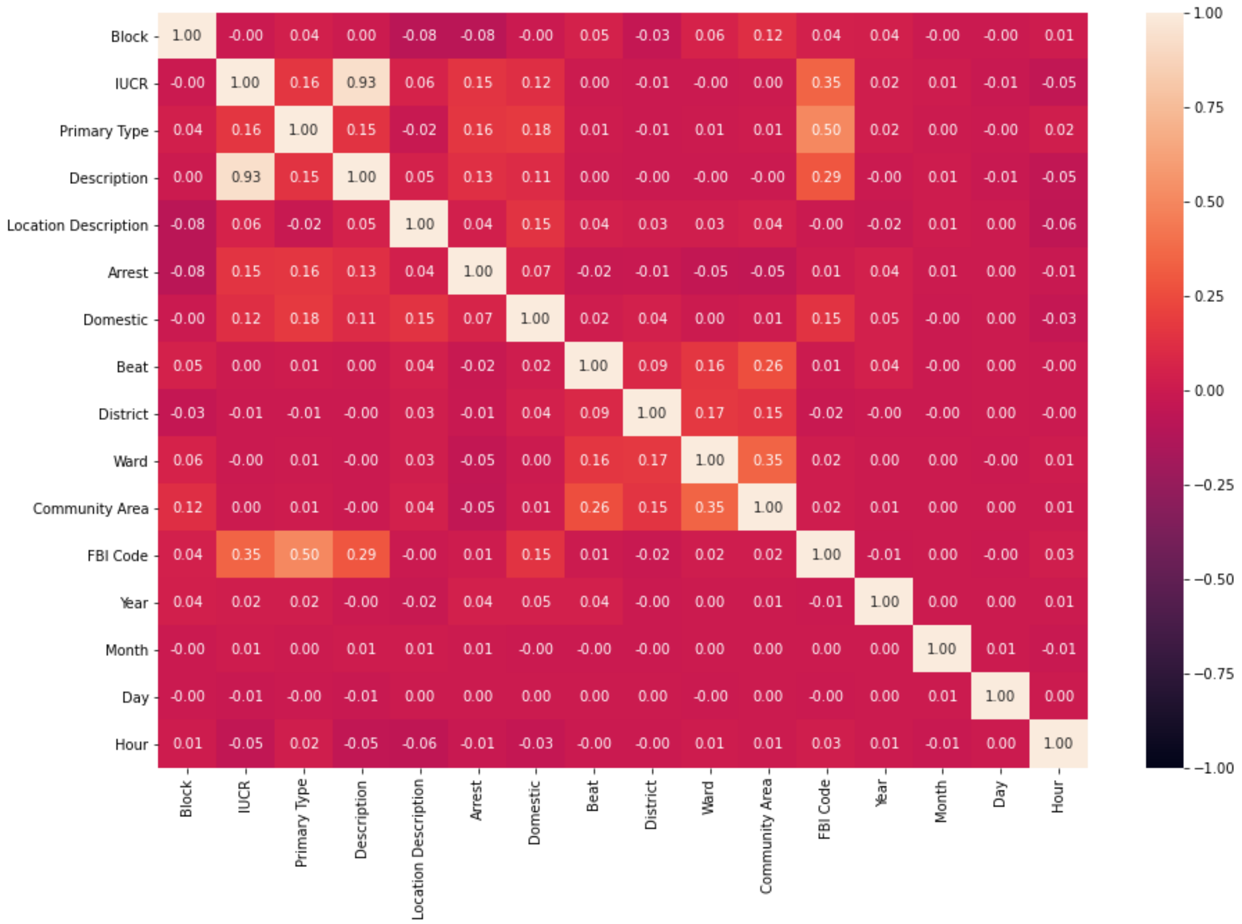


Figure 37. Determining which hour has the most number of specific Primary Types



	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District	Ward	Community Area	FBI Code	Year	Month	Day	Hour
Block	1.00	-0.00	0.04	0.00	-0.08	-0.08	-0.00	0.05	-0.03	0.06	0.12	0.04	0.04	-0.00	-0.00	0.01
IUCR	-0.00	1.00	0.16	0.93	0.06	0.15	0.12	0.00	-0.01	-0.00	0.00	0.35	0.02	0.01	-0.01	-0.05
Primary Type	0.04	0.16	1.00	0.15	-0.02	0.16	0.18	0.01	-0.01	0.01	0.01	0.50	0.02	0.00	-0.00	0.02
Description	0.00	0.93	0.15	1.00	0.05	0.13	0.11	0.00	-0.00	-0.00	-0.00	0.29	-0.00	0.01	-0.01	-0.05
Location Description	-0.08	0.06	-0.02	0.05	1.00	0.04	0.15	0.04	0.03	0.03	0.04	-0.00	-0.02	0.01	0.00	-0.06
Arrest	-0.08	0.15	0.16	0.13	0.04	1.00	0.07	-0.02	-0.01	-0.05	-0.05	0.01	0.04	0.01	0.00	-0.01
Domestic	-0.00	0.12	0.18	0.11	0.15	0.07	1.00	0.02	0.04	0.00	0.01	0.15	0.05	-0.00	0.00	-0.03
Beat	0.05	0.00	0.01	0.00	0.04	-0.02	0.02	1.00	0.09	0.16	0.26	0.01	0.04	-0.00	0.00	-0.00
District	-0.03	-0.01	-0.01	-0.00	0.03	-0.01	0.04	0.09	1.00	0.17	0.15	-0.02	-0.00	-0.00	0.00	-0.00
Ward	0.06	-0.00	0.01	-0.00	0.03	-0.05	0.00	0.16	0.17	1.00	0.35	0.02	0.00	0.00	-0.00	0.01
Community Area	0.12	0.00	0.01	-0.00	0.04	-0.05	0.01	0.26	0.15	0.35	1.00	0.02	0.01	0.00	0.00	0.01
FBI Code	0.04	0.35	0.50	0.29	-0.00	0.01	0.15	0.01	-0.02	0.02	0.02	1.00	-0.01	0.00	-0.00	0.03
Year	0.04	0.02	0.02	-0.00	-0.02	0.04	0.05	0.04	-0.00	0.00	0.01	-0.01	1.00	0.00	0.00	0.01
Month	-0.00	0.01	0.00	0.01	0.01	0.01	-0.00	-0.00	-0.00	0.00	0.00	0.00	0.00	1.00	0.01	-0.01
Day	-0.00	-0.01	-0.00	-0.01	0.00	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00	0.01	1.00	0.00
Hour	0.01	-0.05	0.02	-0.05	-0.06	-0.01	-0.03	-0.00	-0.00	0.01	0.01	0.03	0.01	-0.01	0.00	1.00

Figure 38. Correlation Heatmap for the CC dataset

NCRP dataset

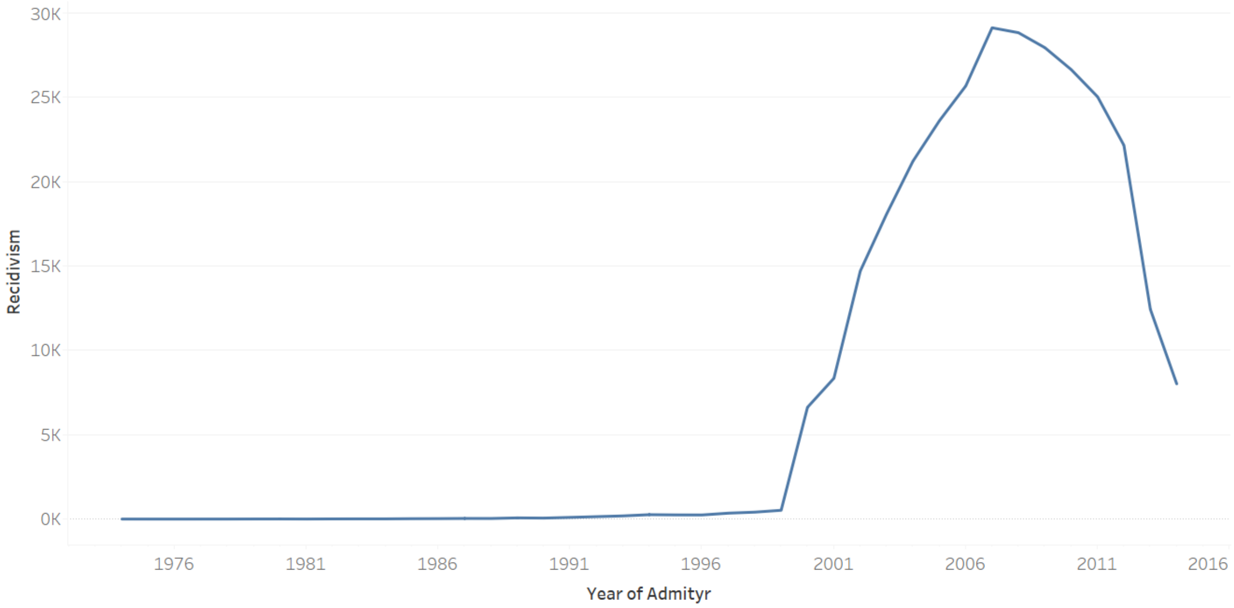


Figure 39. No. of recidivism cases against year of admission

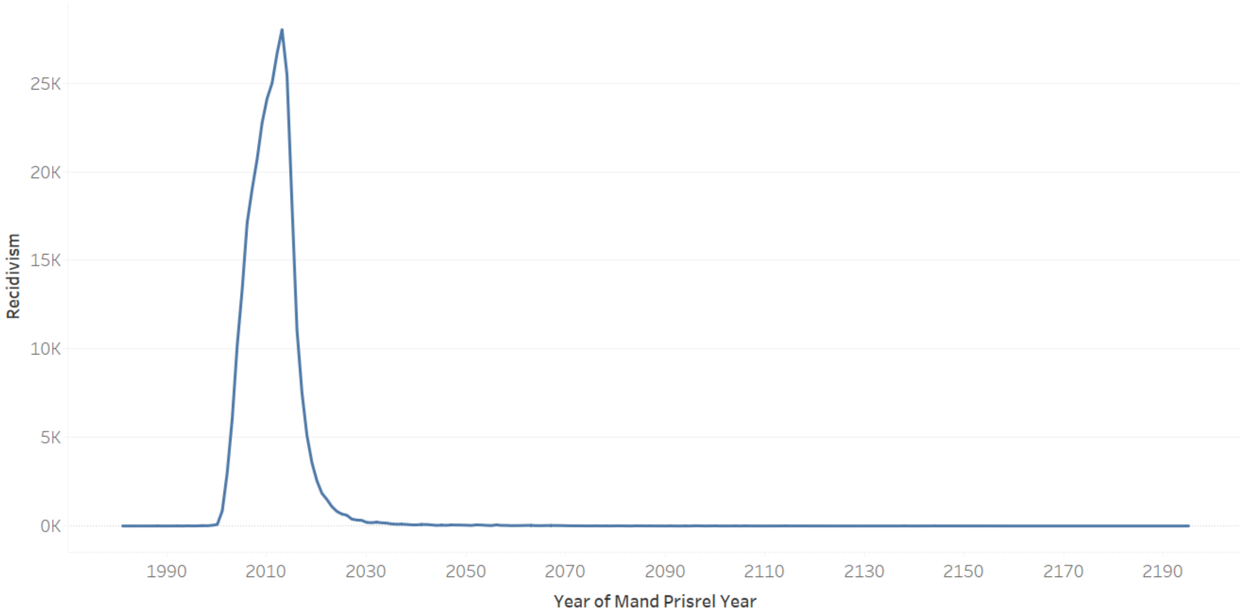


Figure 40. No. of recidivism cases against year of mandatory prison release

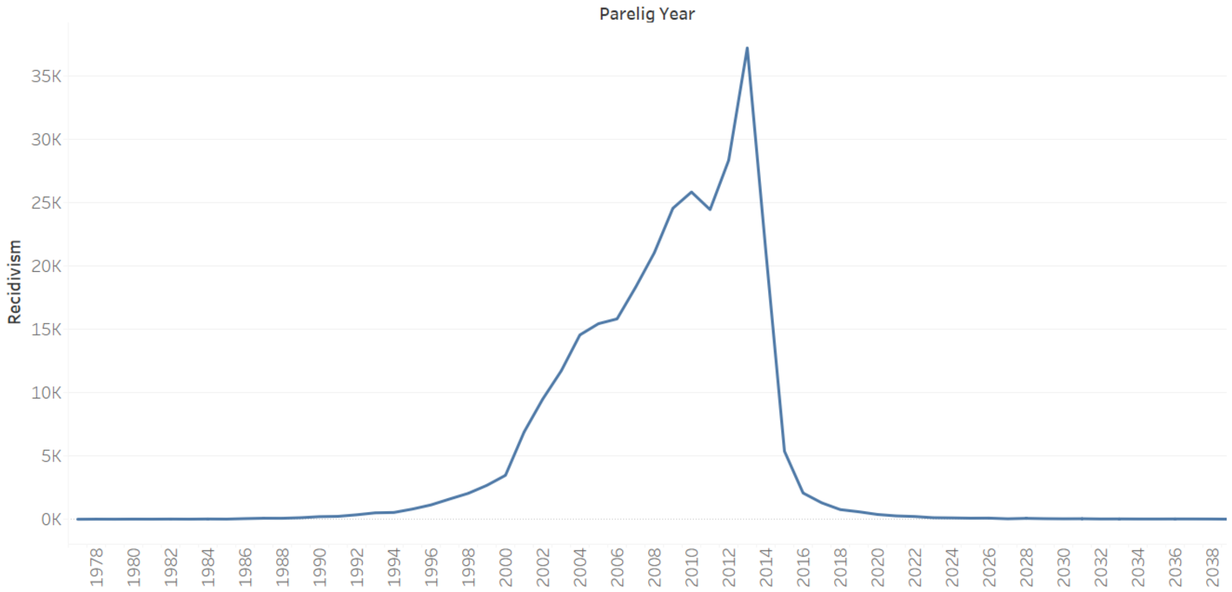


Figure 41. No. of recidivism cases against year of parole eligibility

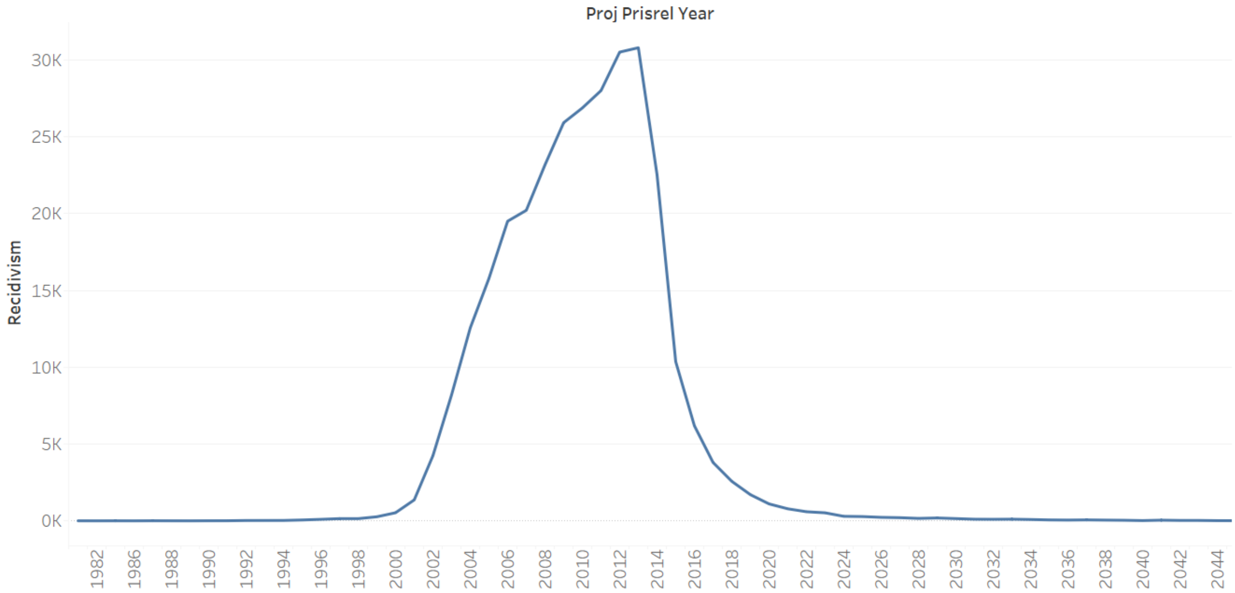


Figure 42. No. of recidivism cases against year of projected prison release

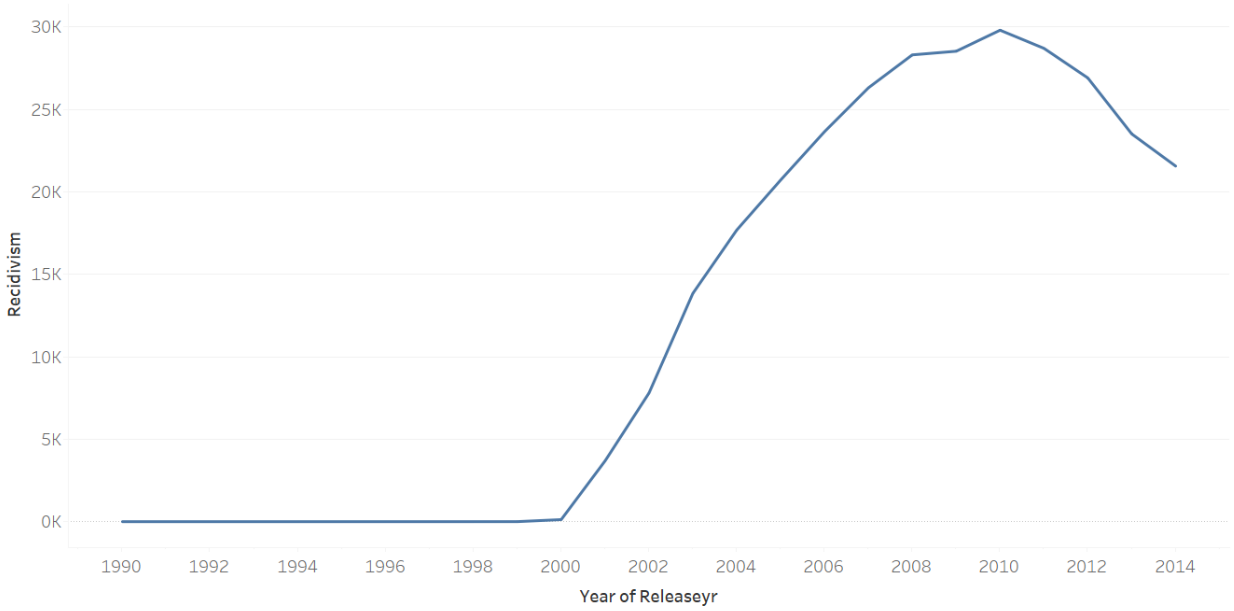


Figure 43. No. of recidivism cases against year of release

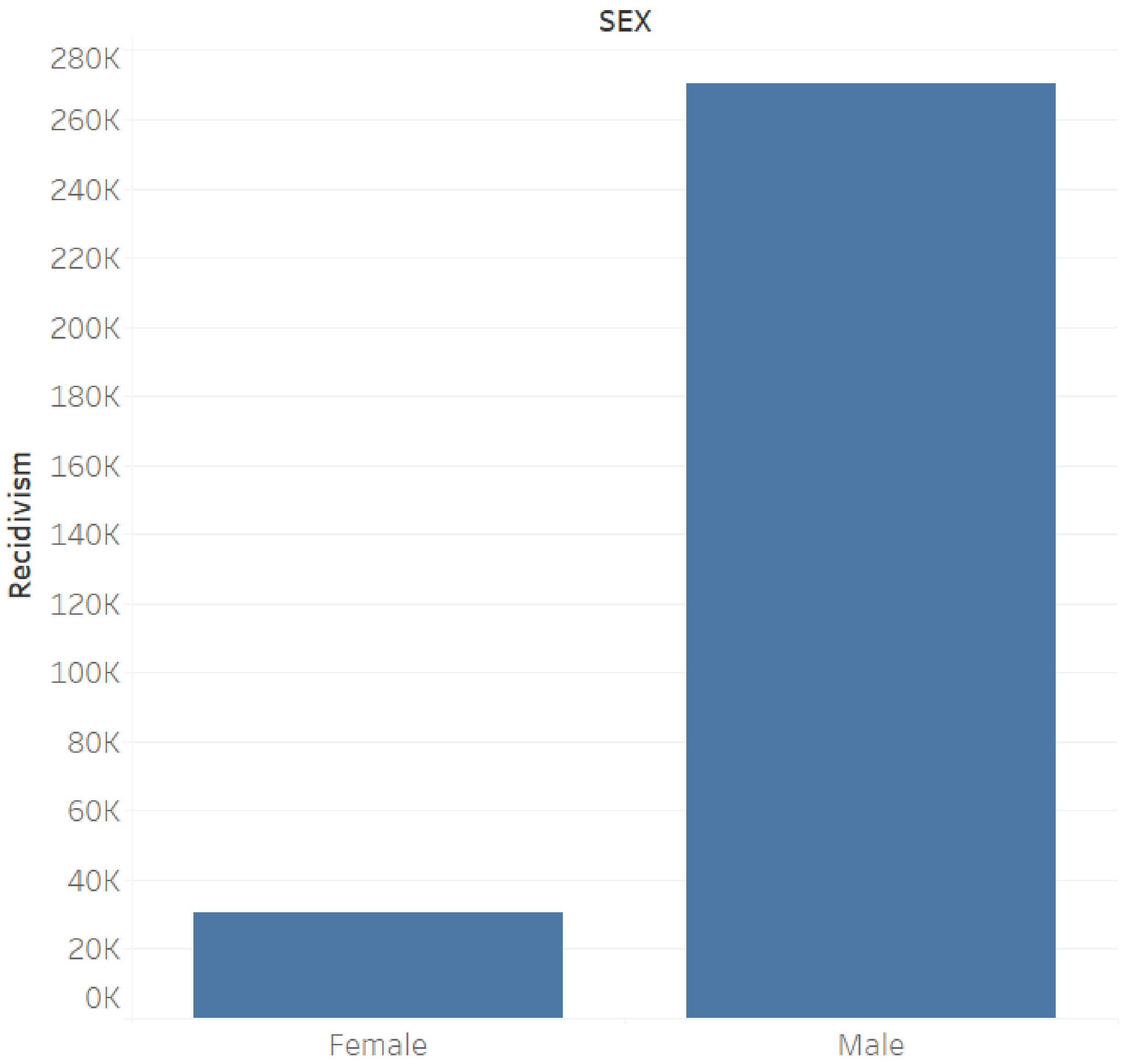


Figure 44. No. of recidivism cases against sex

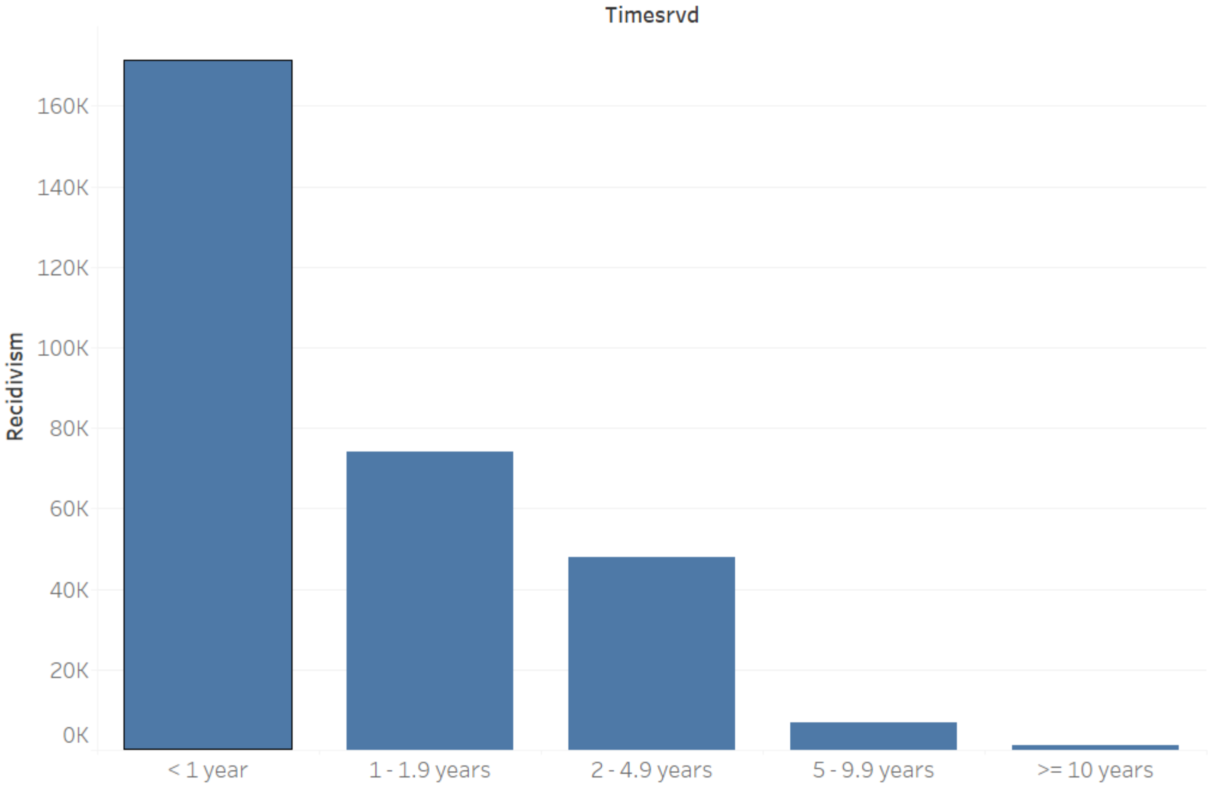


Figure 45. No. of recidivism cases against time served by inmate

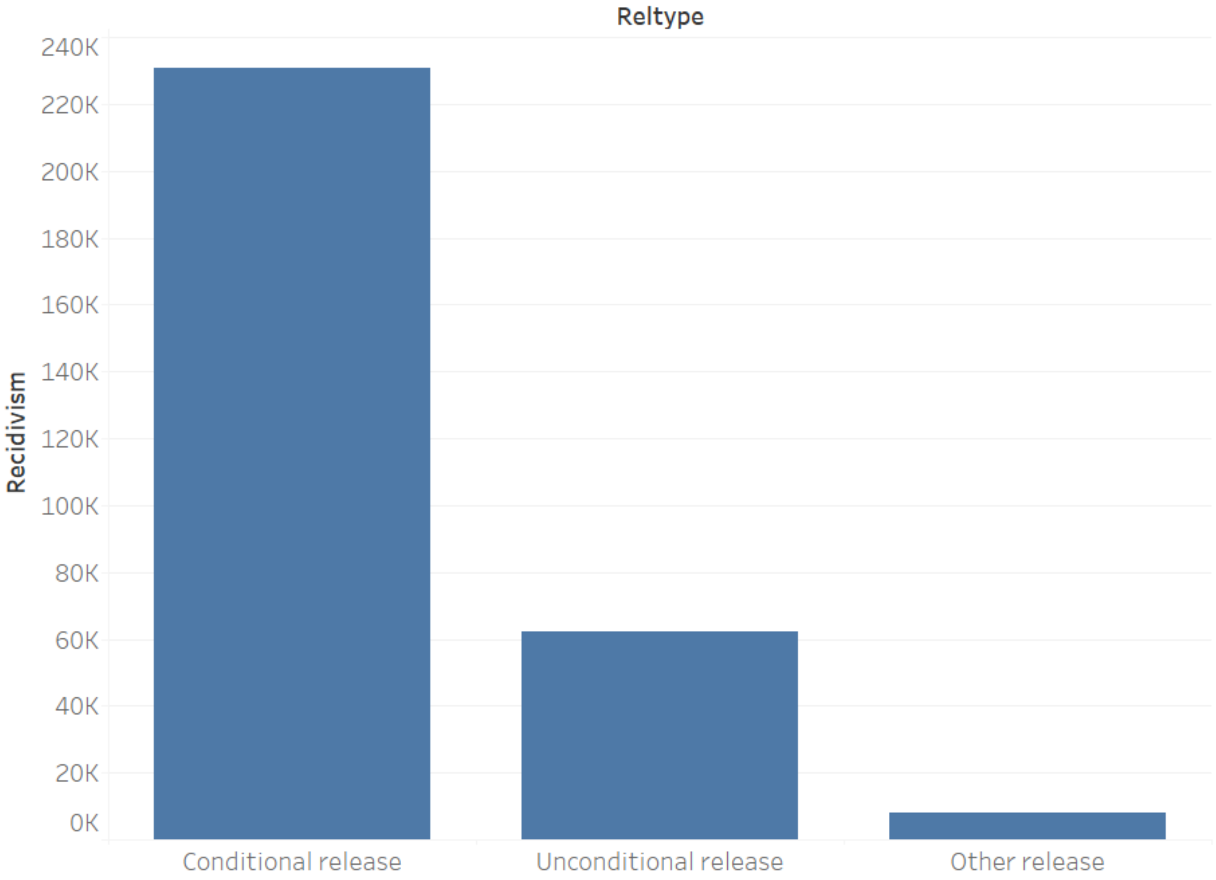


Figure 46. No. of recidivism cases against type of prison release

9.5 Appendix E: Model Code and Results

CC Dataset

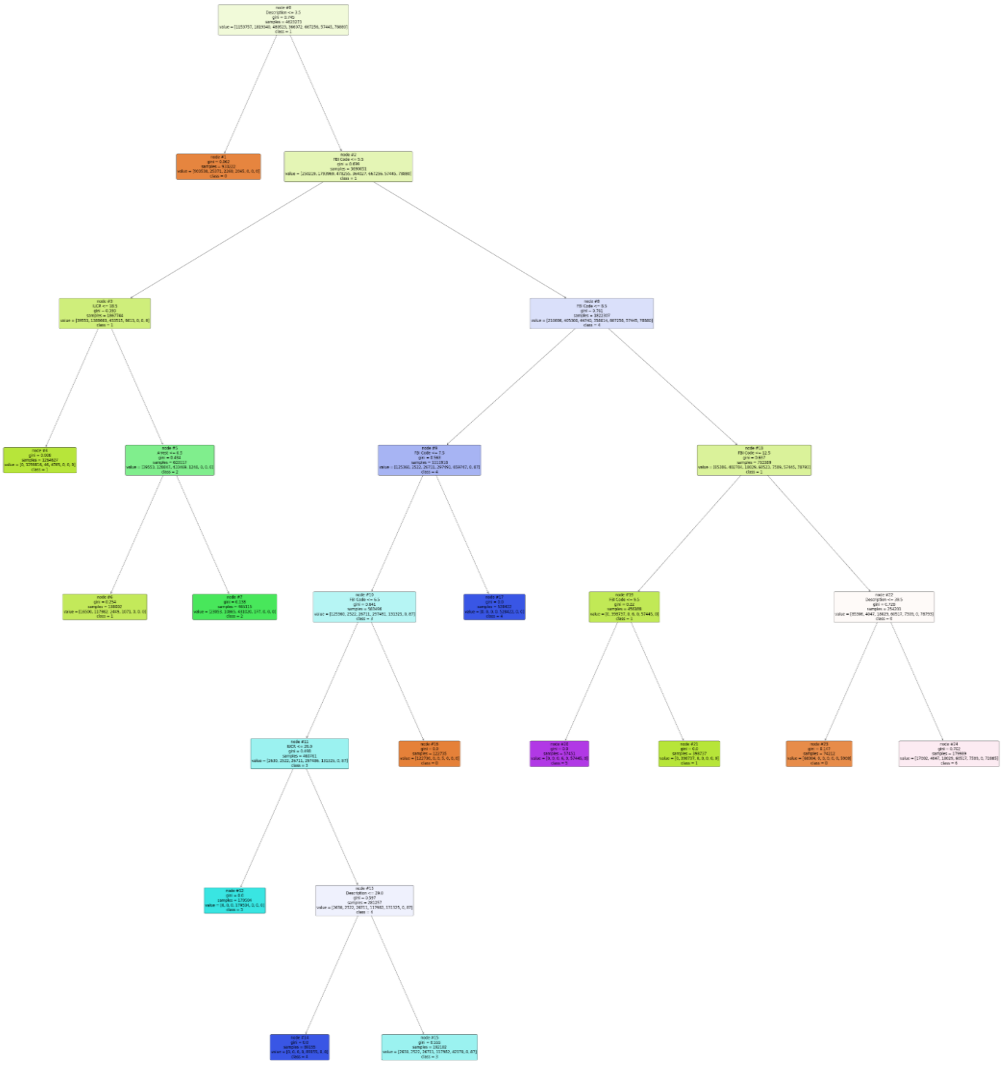


Figure 47. Classification Tree for CART model

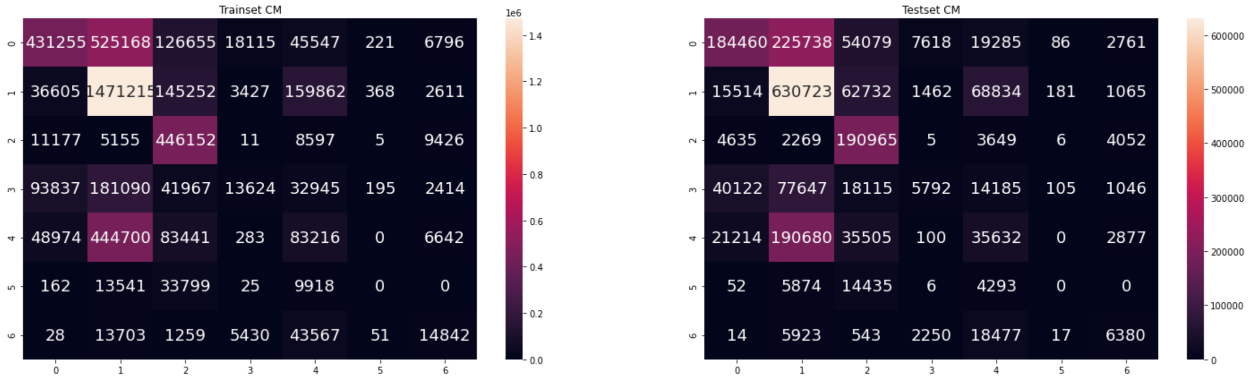
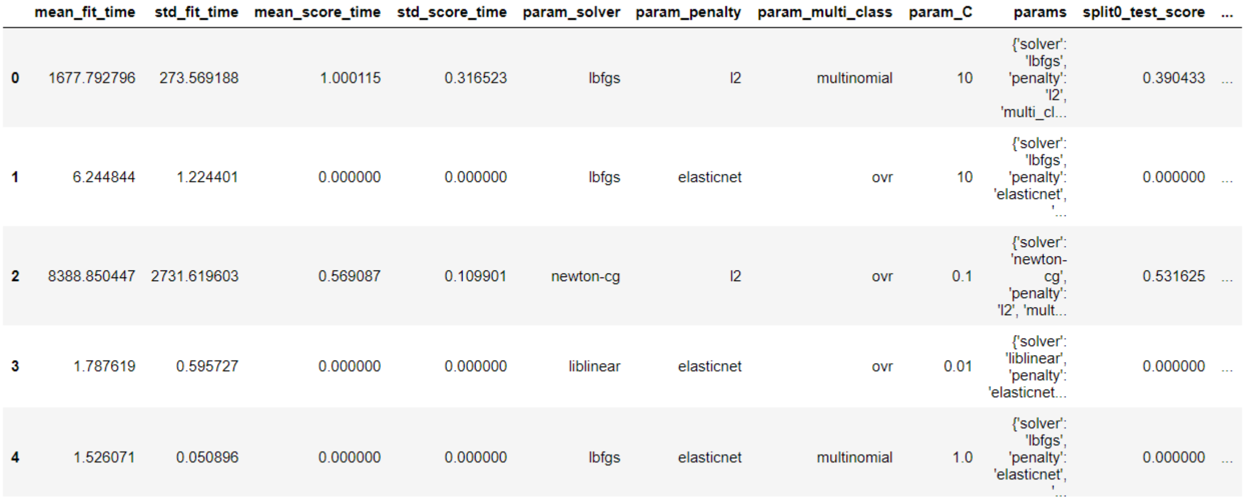
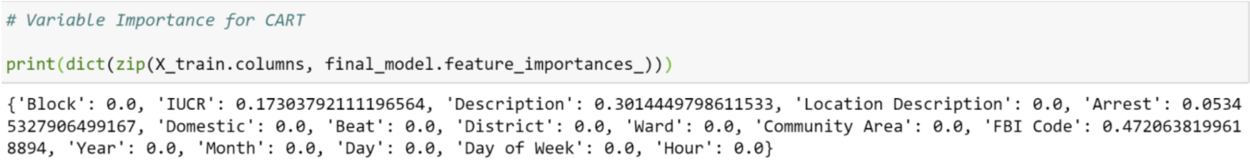


Figure 48. Confusion Matrix for Logistic Regression model



	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_solver	param_penalty	param_multi_class	param_C	params	split0_test_score	...
0	1677.792796	273.569188	1.000115	0.316523	lbfgs	l2	multinomial	10	{'solver': 'lbfgs', 'penalty': 'l2', 'multi_cl...	0.390433	...
1	6.244844	1.224401	0.000000	0.000000	lbfgs	elasticnet	ovr	10	{'solver': 'lbfgs', 'penalty': 'elasticnet', '...	0.000000	...
2	8388.850447	2731.619603	0.569087	0.109901	newton-cg	l2	ovr	0.1	{'solver': 'newton-cg', 'penalty': 'l2', 'mult...	0.531625	...
3	1.787619	0.595727	0.000000	0.000000	liblinear	elasticnet	ovr	0.01	{'solver': 'liblinear', 'penalty': 'elasticnet...	0.000000	...
4	1.526071	0.050896	0.000000	0.000000	lbfgs	elasticnet	multinomial	1.0	{'solver': 'lbfgs', 'penalty': 'elasticnet', '...	0.000000	...

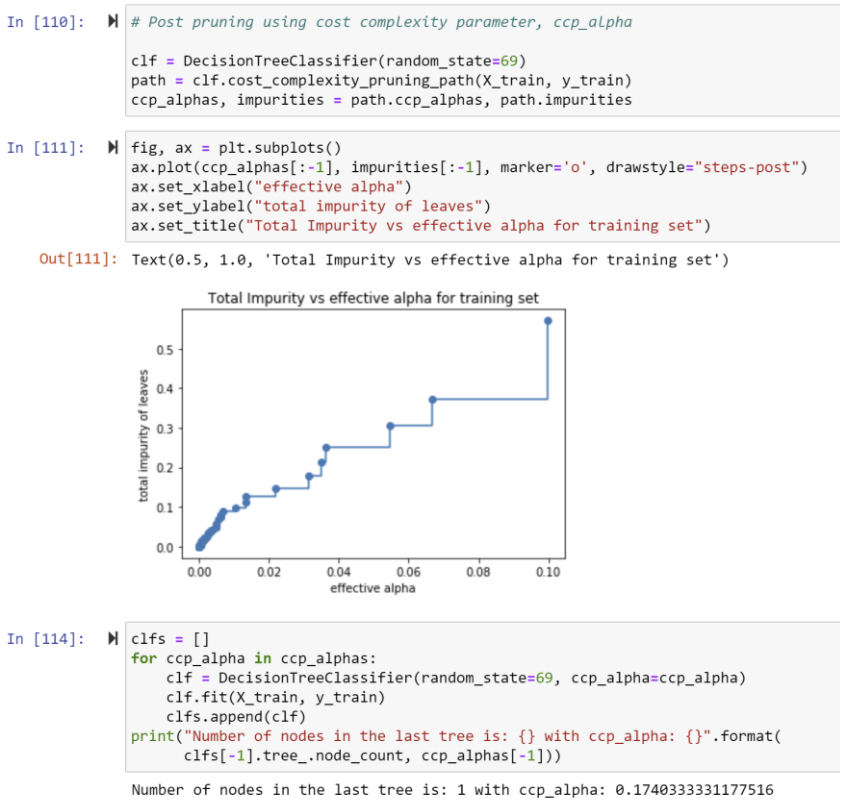
Figure 49. Details of the RandomizedSearchCV results



```
# Variable Importance for CART

print(dict(zip(X_train.columns, final_model.feature_importances_)))
```

```
{'Block': 0.0, 'IUCR': 0.17303792111196564, 'Description': 0.3014449798611533, 'Location Description': 0.0, 'Arrest': 0.0534
5327906499167, 'Domestic': 0.0, 'Beat': 0.0, 'District': 0.0, 'Ward': 0.0, 'Community Area': 0.0, 'FBI Code': 0.472063819961
8894, 'Year': 0.0, 'Month': 0.0, 'Day': 0.0, 'Day of Week': 0.0, 'Hour': 0.0}
```

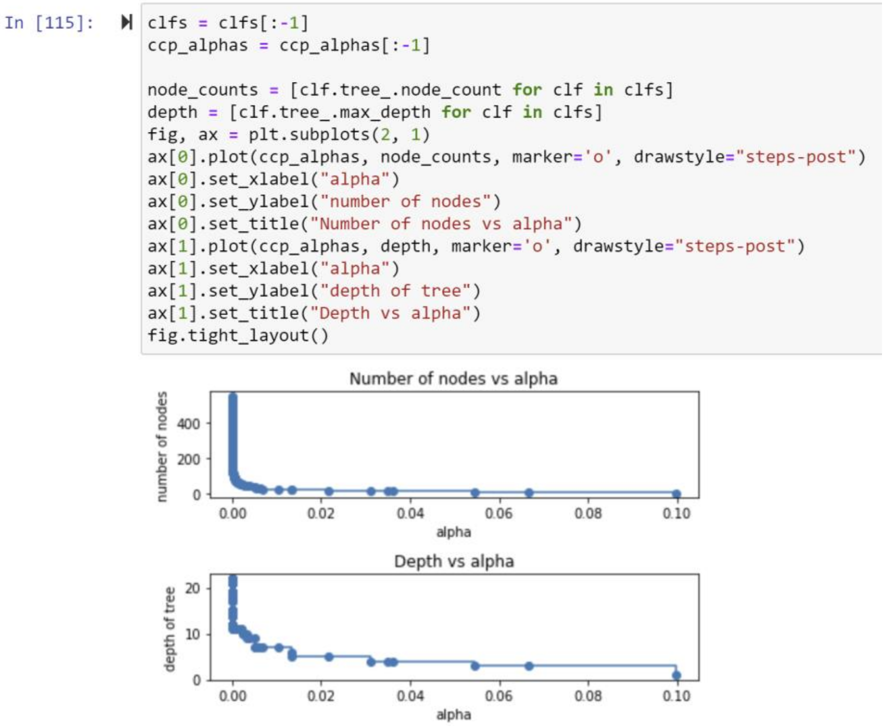
Figure 50. CART Variable Importances



```
In [110]:  # Post pruning using cost complexity parameter, ccp_alpha

            clf = DecisionTreeClassifier(random_state=69)
            path = clf.cost_complexity_pruning_path(X_train, y_train)
            ccp_alphas, impurities = path.ccp_alphas, path.impurities

In [111]:  fig, ax = plt.subplots()
            ax.plot(ccp_alphas[:-1], impurities[:-1], marker='o', drawstyle="steps-post")
            ax.set_xlabel("effective alpha")
            ax.set_ylabel("total impurity of leaves")
            ax.set_title("Total Impurity vs effective alpha for training set")

Out[111]:  Text(0.5, 1.0, 'Total Impurity vs effective alpha for training set')
```

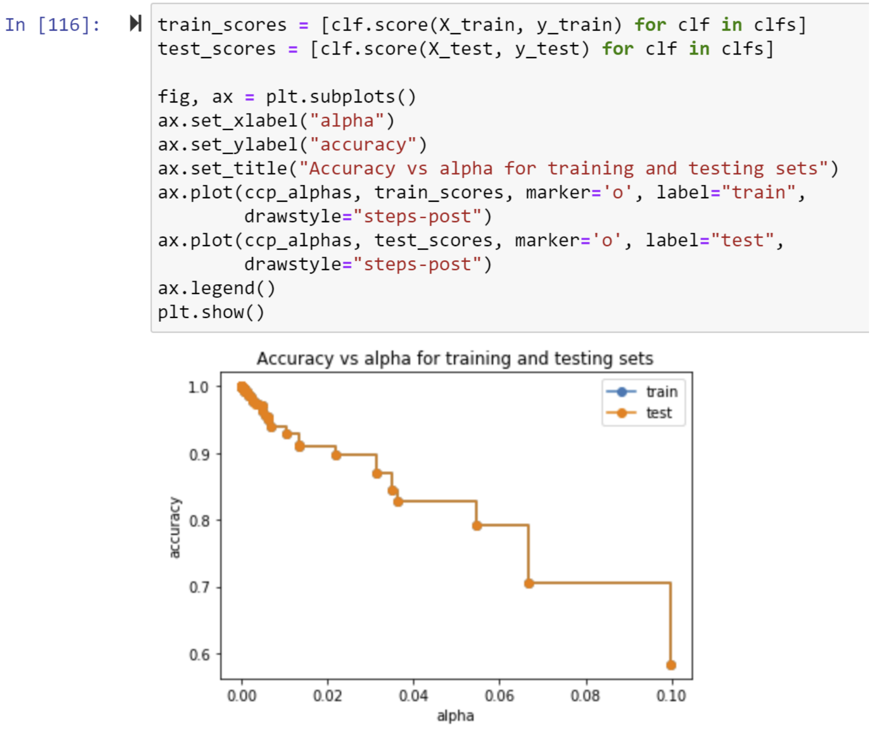
```
In [114]:  clfs = []
            for ccp_alpha in ccp_alphas:
                clf = DecisionTreeClassifier(random_state=69, ccp_alpha=ccp_alpha)
                clf.fit(X_train, y_train)
                clfs.append(clf)
            print("Number of nodes in the last tree is: {} with ccp_alpha: {}".format(
                  clfs[-1].tree_.node_count, ccp_alphas[-1]))

            Number of nodes in the last tree is: 1 with ccp_alpha: 0.1740333331177516
```

Figure 51 (a). CART pruning process



```
In [115]:  clfs = clfs[:-1]
            ccp_alphas = ccp_alphas[:-1]

            node_counts = [clf.tree_.node_count for clf in clfs]
            depth = [clf.tree_.max_depth for clf in clfs]
            fig, ax = plt.subplots(2, 1)
            ax[0].plot(ccp_alphas, node_counts, marker='o', drawstyle="steps-post")
            ax[0].set_xlabel("alpha")
            ax[0].set_ylabel("number of nodes")
            ax[0].set_title("Number of nodes vs alpha")
            ax[1].plot(ccp_alphas, depth, marker='o', drawstyle="steps-post")
            ax[1].set_xlabel("alpha")
            ax[1].set_ylabel("depth of tree")
            ax[1].set_title("Depth vs alpha")
            fig.tight_layout()
```

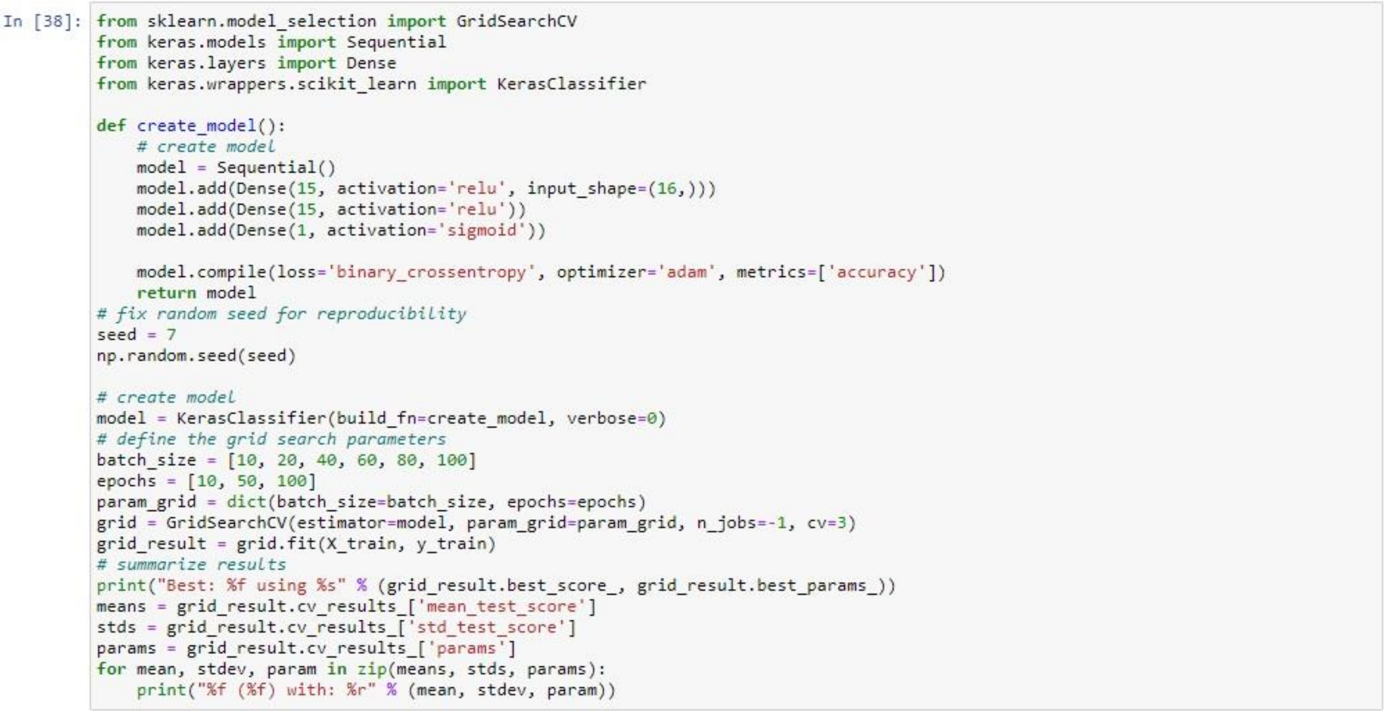
Figure 51 (b). CART pruning process



```
In [116]:  train_scores = [clf.score(X_train, y_train) for clf in clfs]
           test_scores = [clf.score(X_test, y_test) for clf in clfs]

           fig, ax = plt.subplots()
           ax.set_xlabel("alpha")
           ax.set_ylabel("accuracy")
           ax.set_title("Accuracy vs alpha for training and testing sets")
           ax.plot(ccp_alphas, train_scores, marker='o', label="train",
                   drawstyle="steps-post")
           ax.plot(ccp_alphas, test_scores, marker='o', label="test",
                   drawstyle="steps-post")
           ax.legend()
           plt.show()
```

Figure 51 (c). CART pruning process

NCRP Dataset



```
In [38]: from sklearn.model_selection import GridSearchCV
         from keras.models import Sequential
         from keras.layers import Dense
         from keras.wrappers.scikit_learn import KerasClassifier

         def create_model():
             # create model
             model = Sequential()
             model.add(Dense(15, activation='relu', input_shape=(16,)))
             model.add(Dense(15, activation='relu'))
             model.add(Dense(1, activation='sigmoid'))

             model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
             return model
         # fix random seed for reproducibility
         seed = 7
         np.random.seed(seed)

         # create model
         model = KerasClassifier(build_fn=create_model, verbose=0)
         # define the grid search parameters
         batch_size = [10, 20, 40, 60, 80, 100]
         epochs = [10, 50, 100]
         param_grid = dict(batch_size=batch_size, epochs=epochs)
         grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
         grid_result = grid.fit(X_train, y_train)
         # summarize results
         print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
         means = grid_result.cv_results_['mean_test_score']
         stds = grid_result.cv_results_['std_test_score']
         params = grid_result.cv_results_['params']
         for mean, stdev, param in zip(means, stds, params):
             print("%f (%f) with: %r" % (mean, stdev, param))
```

Figure 52. Code snippet to determine batch and epoch size code

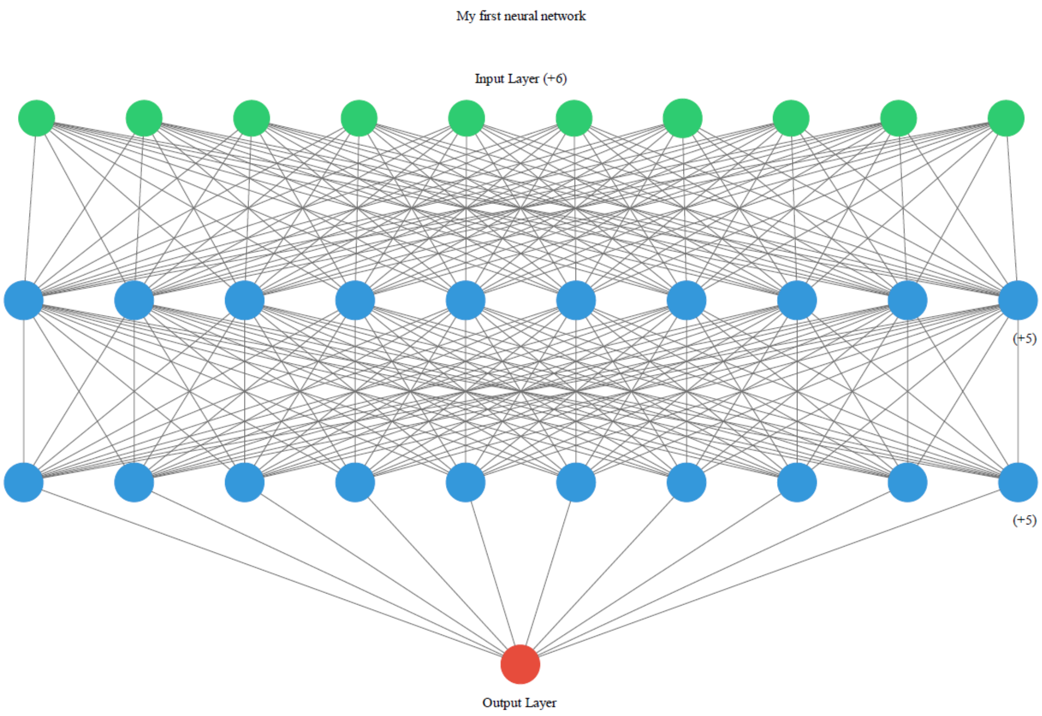


Figure 53. Visual Representation of the NN model

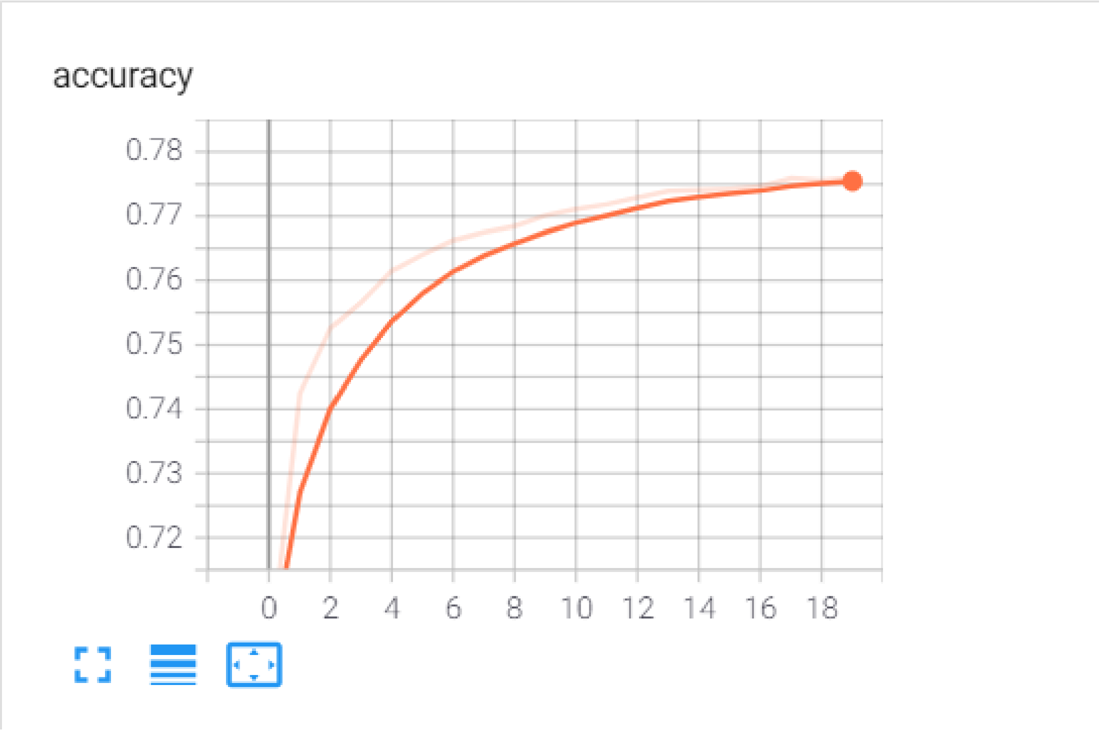


Figure 54. Graphical Representation of the NN model from TensorBoard

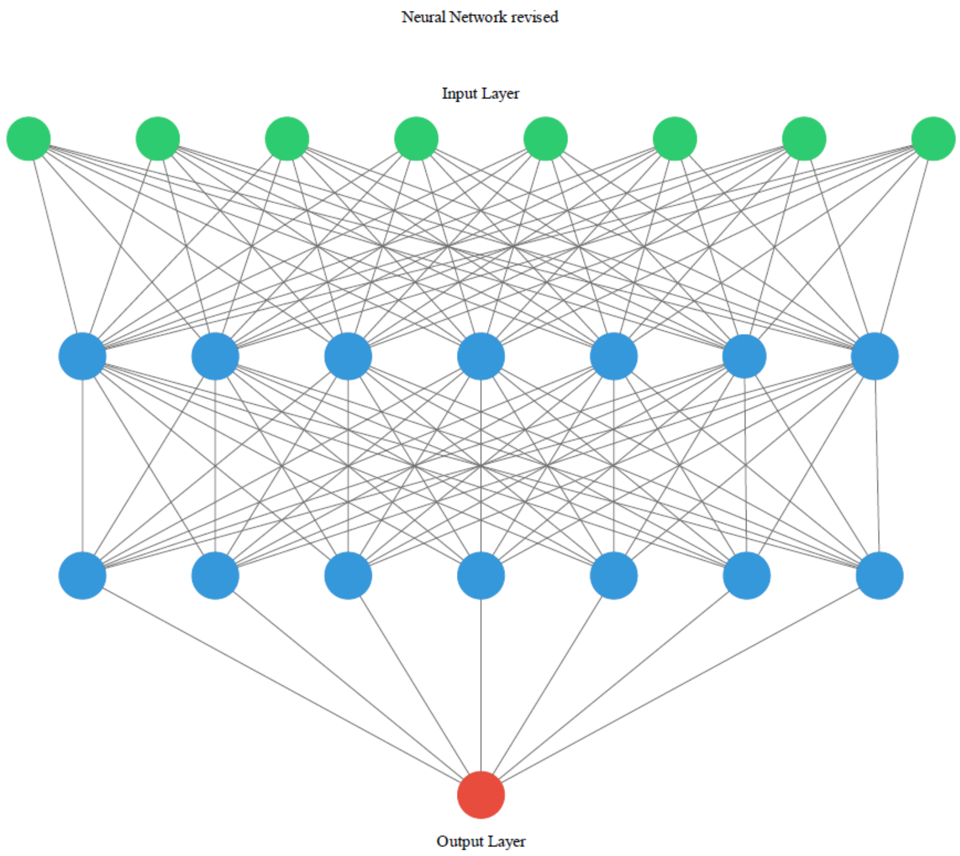


Figure 55. Visual Representation of the new NN model

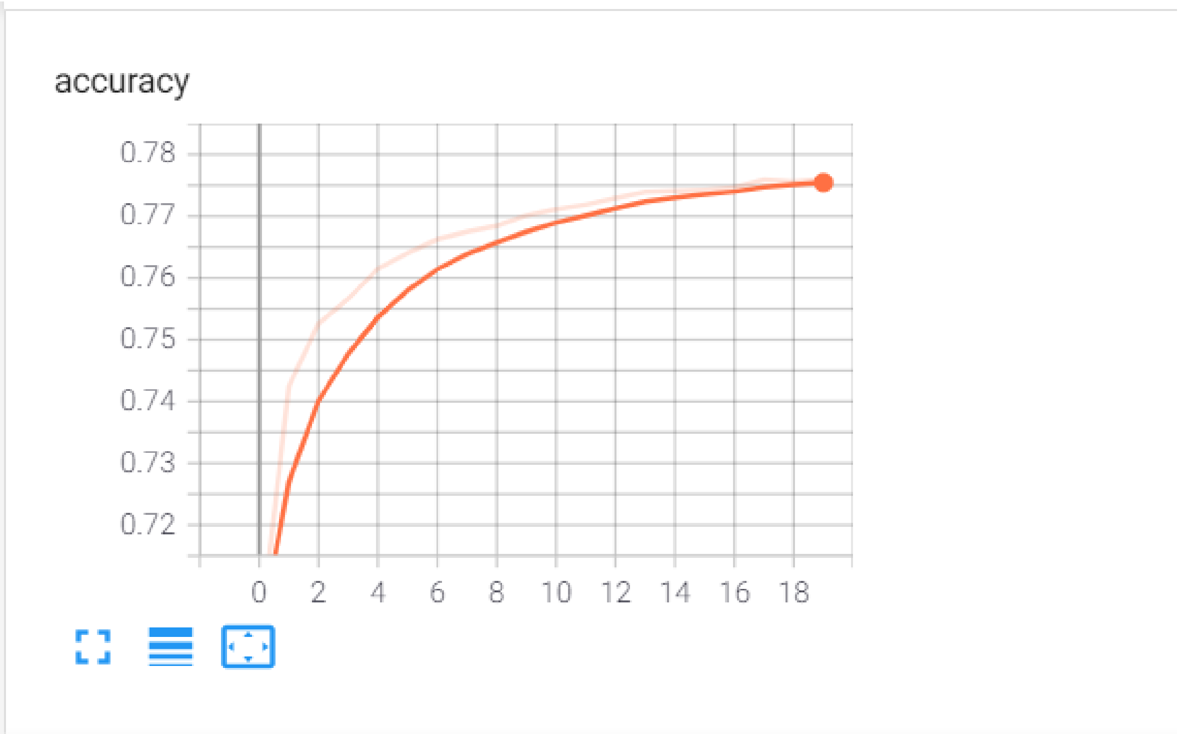


Figure 56. Graphical Representation of the new NN model from TensorBoard

9.6 Appendix F: Data Exploration for Singapore Context

On exploration of another available dataset, zooming in on the significant ‘Cheating & Related’ offences, we can see that the majority of ‘Cheating & Related’ offenders are above 21 years old, however, the youngest offenders can be as young as 7 years old.

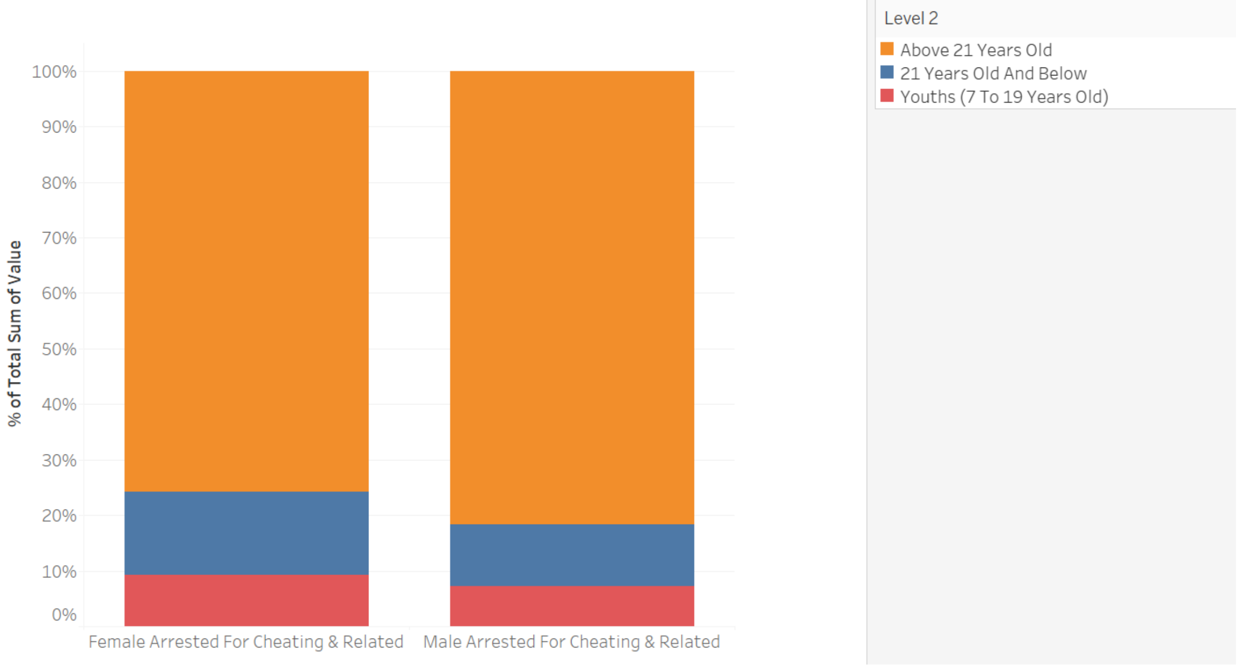
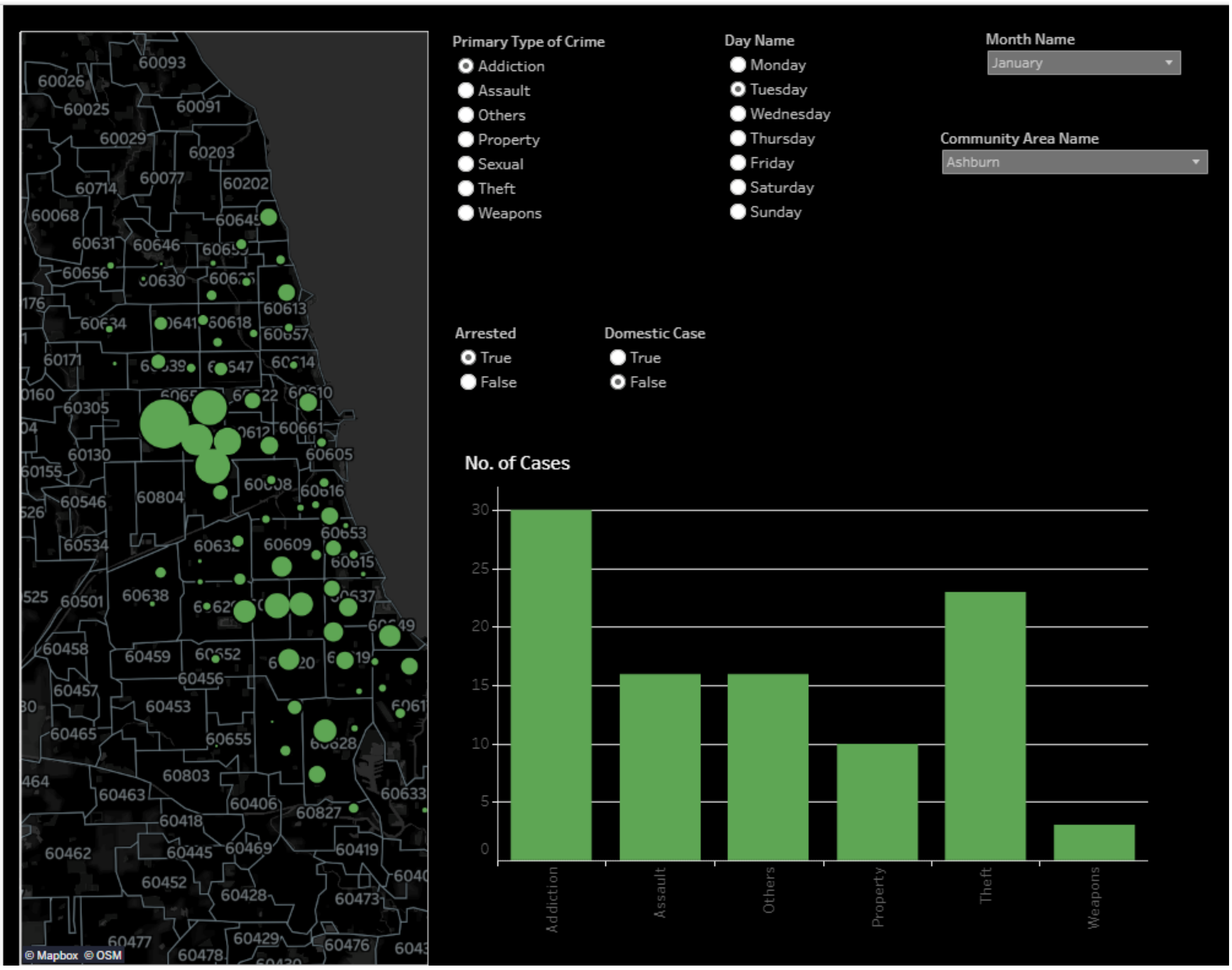


Figure 57. ‘Cheating & Related’ offences between genders and age groups

9.7 Appendix G: Dashboard Interface



Primary Type of Crime
● Addiction
○ Assault
○ Others
○ Property
○ Sexual
○ Theft
○ Weapons

Day Name
○ Monday
● Tuesday
○ Wednesday
○ Thursday
○ Friday
○ Saturday
○ Sunday

Month Name
January

Community Area Name
Ashburn

Arrested
● True
○ False

Domestic Case
○ True
● False

Figure 58. Dashboard for visualising results of CC dataset model

10 References

1. Ananyan, Sergei, "Crime pattern analysis through text mining" (2004). AMCIS 2004 Proceedings. 236. http://aisel.aisnet.org/amcis2004/236
2. Bachner, J., Ms. (2013). Retrieved from http://www.businessofgovernment.org/sites/default/files/Management%20Predictive%20Policing.pdf
3. Brownlee, J. (2019, August 06). How to configure the number of layers and nodes in a neural network. Retrieved March 26, 2021, from https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/
4. Brownlee, J. (2019, October 25). Difference between a batch and an epoch in a neural network. Retrieved March 26, 2021, from https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/
5. Brownlee, J. (2020, August 10). Ensemble learning algorithm complexity and Occam's razor. Retrieved April 03, 2021, from https://machinelearningmastery.com/ensemble-learning-and-occams-razor/#:~:text=Occam's%20razor%20is%20a%20heuristic,false%20and%20should%20be%20abandoned.
6. Brownlee, J. (2020, August 20). How to choose a feature selection method for machine learning. Retrieved March 26, 2021, from https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
7. Brownlee, J. (2020, August 30). Multi-Label classification with deep learning. Retrieved April 01, 2021, from https://machinelearningmastery.com/multi-label-classification-with-deep-learning/
8. Bunge, M. (2006). A systemic perspective on crime. In P. Wikström & R. Sampson (Eds.), *The Explanation of Crime: Context, Mechanisms and Development* (Pathways in Crime, pp. 8-30). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511489341.002
9. Clark, M., Mr. (2019, May 3). Long-Term Recidivism Studies Show High Arrest Rates. Retrieved February 24, 2021, from
10. Clark, T. (2020, July 11). Why neural nets can approximate any function. Retrieved March 26, 2021, from https://towardsdatascience.com/why-neural-nets-can-approximate-any-function-a878768502f0
11. Crime rate by country 2021. (n.d.). Retrieved April 01, 2021, from https://worldpopulationreview.com/country-rankings/crime-rate-by-country
12. Crime up more than 11% in first half of 2020, mainly due to rise in scam cases. (2020, August 26). Retrieved March 30, 2021, from https://www.channelnewsasia.com/news/singapore/crime-rate-statistics-first-half-2020-online-scams-13053746
13. Dunnett, S., Ms, Leigh, J., Ms, & Jackson, L., Ms. (2018, February 22). Optimising police dispatch for incident response in real time. Retrieved February 23, 2021, from https://www.tandfonline.com/doi/full/10.1080/01605682.2018.1434401
14. Glorfeld, J. (2018, October 26). Homicide, burglary influenced by weather, time of day. Retrieved from Cosmos: https://cosmosmagazine.com/society/homicide-burglary-influenced-by-weather-time-of-day
15. Hajela, G., Chawla, M., & Rasool, A. (2020, April 16). A clustering based hotspot identification approach for crime prediction. Retrieved March 31, 2021, from https://www.sciencedirect.com/science/article/pii/S1877050920308231
16. Jayaweera et al. (2015). Crime Analytics: Analysis of Crimes Through Newspaper Articles. https://www.researchgate.net/publication/274956334_Crime_Analytics_Analysis_of_Crimes_Through_Newspaper_Articles

17. Koehrsen, W. (2018, January 10). Hyperparameter tuning the random forest in Python. Retrieved April 02, 2021, from https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
18. Kong, Y., & Yu, T. (2018, November 07). A deep neural network model using random forest to Extract feature representation for gene expression data classification. Retrieved March 26, 2021, from https://www.nature.com/articles/s41598-018-34833-6#citeas
19. Lyon, E., Mr. (2019, February 5). Retrieved February 23, 2021, from https://www.prisonlegalnews.org/news/2019/feb/5/illinois-calculates-high-costs-recidivism/#:~:text=It%20is%20anticipated%20that%2096,reoffenders%20headed%20back%20to%20prison.
20. Mahmud, A. (2021, February 05). Recidivism rate at all-time low; more inmates serving part of jail term in community: Prison service. Retrieved March 30, 2021, from https://www.channelnewsasia.com/news/singapore/prison-recidivism-inmate-community-based-programme-sps-14110318
21. Matthew Ranson (2014), "Crime, weather, and climate change" Retrieved April 03, 2021, from https://www.sciencedirect.com/science/article/abs/pii/S0095069613001289#!
22. Nooney, K. (2019, February 12). Deep dive into multi-label classification..! (with detailed case study). Retrieved April 01, 2021, from https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff
23. Robbins, B. (2017, July 14). Machine learning: How black is this beautiful black box. Retrieved March 28, 2021, from https://towardsdatascience.com/machine-learning-how-black-is-this-black-box-f11e4031fdf
24. Robert P. Shumate, Richard F. Crowther, Quantitative Methods for Optimizing the Allocation of Police Resources, 57 J. Crim. L. Criminology & Police Sci. 197 (1966)
25. Rongbin Xu, Xiuqin Xiong, Michael J. Abramson, Shanshan Li, Yuming Guo (2021), "Association between ambient temperature and sex offense: A case-crossover study in seven large US cities, 2007–2017", Retrieved April 03, 2021 from https://www.sciencedirect.com/science/article/abs/pii/S2210670721001189
26. Sharma, S., Mr. (2019, February 14). Activation functions in neural networks. Retrieved March 26, 2021, from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6
27. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.