Geographical Analysis and Prediction of Chicago Crime

Wang Peinan

Introduction

Crime is a major social problem that every country needs to face, affecting public safety, child development and the socio-economic status of adults. My interest in this aspect of research was renewed when I heard the news of the tragic death of our alumnus, Shaoxiong Zheng, near the University of Chicago. In this project I have analyzed the trends of crime based on time and location, in addition I have used supervised machine learning techniques including decision trees, random forests and KNN, to predict crime based on time, location and other parameters. I am confident that this project will help reduce crime rates, improve public safety and alleviate citizens' fears and insecurities.

Code Description

In this project, I used Python to process the data and do predictive modelling analysis. In addition, I use RStudio to do data visualization and exploratory data analysis. For detailed code see Jupyter Notebook and R markdown in the appendix.

Datasets

Crime data of Chicago (from 2001 to 2024) obtained from the City of Chicago data portal. This dataset contains about 8 million Chicago crime incidents from 2001 to 2024. The variables include crime date, type, description, location, etc. For a detailed explanation of the variables used in this project see the "Data Overview" section.
Link: https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/about_data

Data Processing

I started by removing missing and null values, which accounted for less than 1% of the data, and filtering out irrelevant features in the dataset. Then I reduced the number of crime types by merging some similar crime types. For example, I combined SEX OFFENSE and CRIM SEXUAL ASSAULT in one category. I chose data from 2014 to 2023 as the accuracy stabilized for this time period.

In the Uniform Crime Reporting (UCR) system, crimes are categorized as Part I and Part II. Part I crimes include murder, rape, robbery and aggravated assault, which are violent crimes. Burglary, grand larceny, auto theft and arson, which are property crimes. Part II crimes include simple assaults, petit larceny, gambling, alcoholism, etc. Overall, Part I are the crimes that are considered more serious while Part II are the crimes that are considered less serious. Since the UCR classification is based primarily on the severity and nature of the crime (based primarily on the type of crime itself), I have reclassified all crime types in the dataset by "IUCR" (i.e., Illinois Uniform Crime Reporting Code) into two categories, Part I and Part II.
Link: https://en.wikipedia.org/wiki/Uniform_Crime_Reports

Data Overview

Table 1 is a preview of my data after pre-processing. I used the "DataFrame" data structure from the "Pandas" library for processing and presenting the data. "Block" is the partially redacted address where the incident occurred, placing it on the same block as the actual address. "Primary Type" is the primary description of the IUCR code. "District" indicates the police district where the incident occurred. "Ward" is the ward (City Council district) where the incident occurred. "Community Area" indicates the community area where the incident occurred, and Chicago has 77 community areas in total. "Year" indicates the year when the incident occurred. "Latitude" and "longitude" refer to the specific coordinates at which the incident occurred. "UCR_PART" refers to the specific category of the incident according to the UCR classification criteria. "Month" indicates the month when the incident occurred. "Day" indicates the date when the incident occurred. "Time" indicates the time period in which the incident occurred. I customized four time periods according to the time, namely, early morning means 1 a.m. to 7 a.m., late morning means 8 a.m. to 1 p.m., afternoon means 2 p.m. to 7 p.m., and night is 8 p.m. to 12 a.m. "Weekday" indicates the weekday when incident occurred.

Tab. 1 Overview of the dataset

	Block	Primary Type	District	Ward	Community Area	Year	Latitude	Longitude	UCR_PART	Month	Day	Time	Weekday
0	035XX S INDIANA AVE	THEFT	2.0	3.0	35.0	2020	41.830482	-87.621752	UCR_PART_I	5	7	Late Morning	Thursday
1	005XX W 32ND ST	BATTERY	9.0	11.0	60.0	2020	41.836310	-87.639624	UCR_PART_I	4	16	Early Morning	Thursday
2	081XX S COLES AVE	ASSAULT	4.0	7.0	46.0	2020	41.747610	-87.549179	UCR_PART_I	7	1	Late Morning	Wednesday
3	065XX S WOLCOTT AVE	BATTERY	7.0	15.0	67.0	2020	41.774878	-87.671375	UCR_PART_I	9	27	Night	Sunday
4	081XX S LOOMIS BLVD	WEAPONS VIOLATION	6.0	21.0	71.0	2020	41.746221	-87.658477	UCR_PART_II	8	4	Night	Tuesday

Data Visualization

I used "ggplot2" library in R and the website www.kepler.gl to generate various visualization plots. Various plots were generated to analyze and gain some insights into the data. "ggplot2" was used to produce graphs based on time and types of crimes committed. The website www.kepler.gl was used to produce graphs based on latitude and longitude of the location where the crimes were committed.

Visualization Based on Time

Figure 1 illustrates the trend of crime types across the past decade. In all years the number of crimes in UCR Part I was substantially greater than the number of crimes in UCR Part II. It is worth noting that in both 2020 and 2021, the number of crimes in both UCR Part I and UCR Part II decreased somewhat. This suggests that environmental factors at that time (possibly the epidemic) contributed to the overall decrease in the number of crimes.

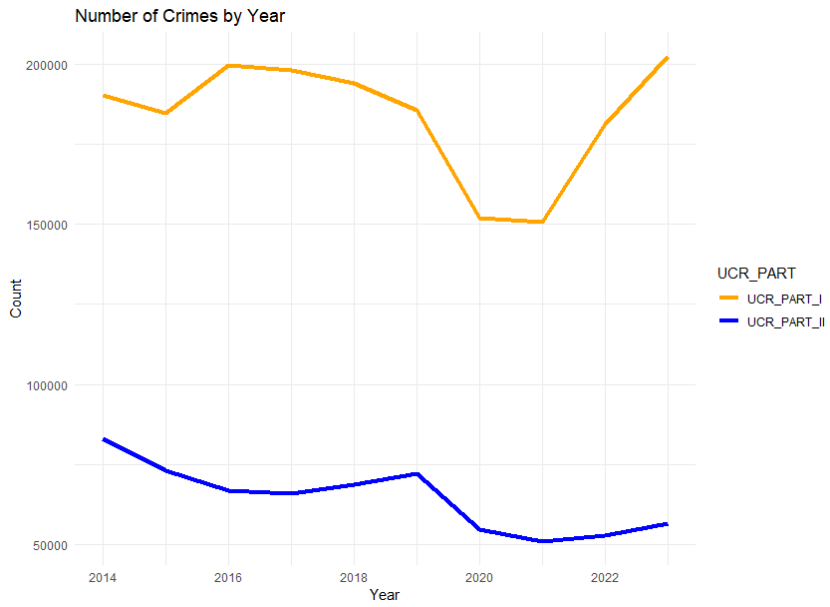


Fig. 1 Trend of crime types across the past decade (2014-2023)

Figure 2 shows the total number of crimes per month. There are no significant peaks or troughs in the number of crimes in UCR Part II. However, the number of crimes in UCR Part I peaks in July and August, which we can say are the most unsafe months.

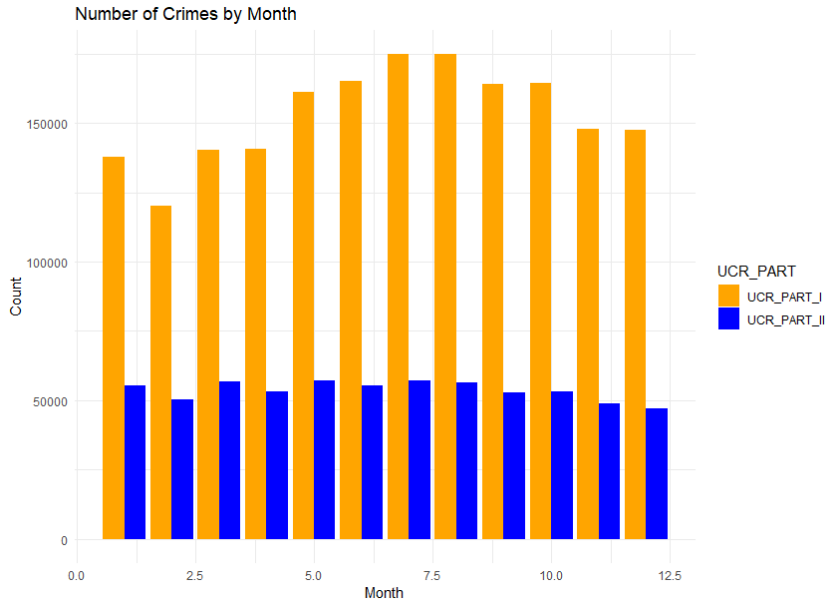


Fig. 2 Number of Crimes by Month (2014-2023)

Figure 3 shows the total number of crimes for each time period. The number of UCR Part I crimes peaks at "Afternoon", indicating that the majority of UCR Part I crimes occur between 2 p.m. and 7 p.m. However, the number of UCR Part II crimes peaks at "Late Morning", indicating that the majority of UCR Part I crimes occur between 8 a.m. and 1 p.m.

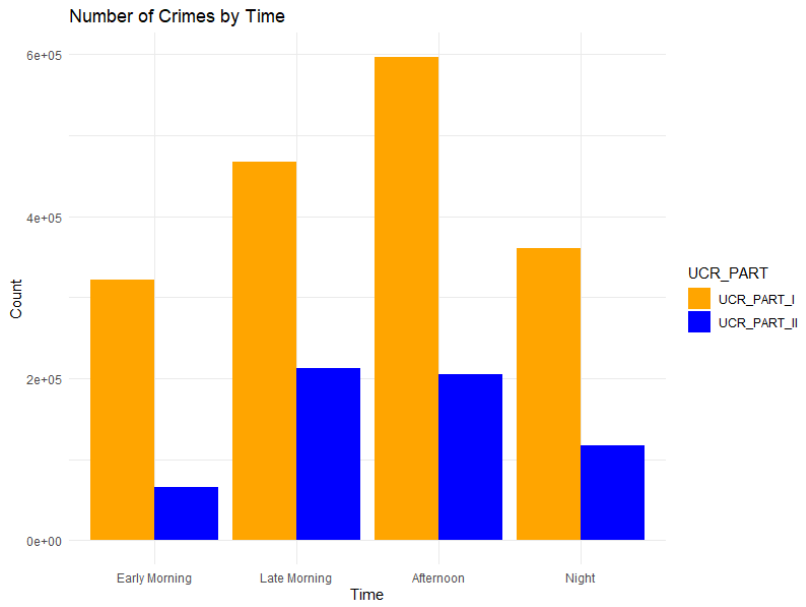


Fig. 3 Number of Crimes by Time (2014-2023)

Visualization Based on Types of Crime

Figure 4 plots the type of crime versus the count of that particular crime. From the graph we can see that the most crimes occurring in Chicago between 2014 and 2023 fall under the UCR Part I, and theft is the type of crime that occurs the most. In addition, the most frequent type of crime in UCR Part II is deceptive practice.

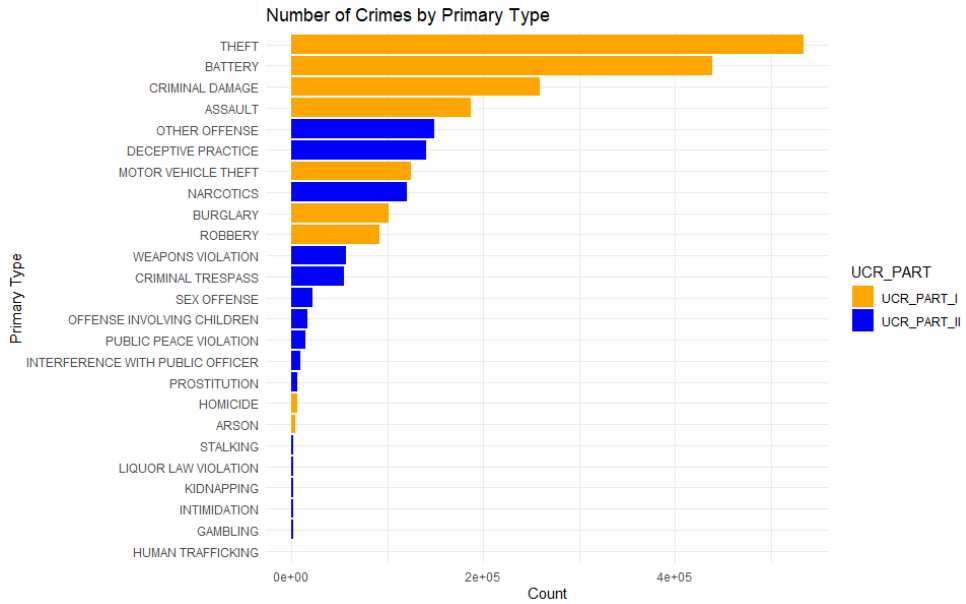


Fig. 4 Types of crime and their UCR PART (2014-2023)

Visualization Based on Location

Figure 5 illustrates the location of the specific coordinates of each crime on the map. From the figure we can see that the crimes occur in almost all areas of Chicago, except for the northwest side area of District 16. In addition, the number of crimes in UCR Part I is significantly larger than the number of crimes in UCR Part II.

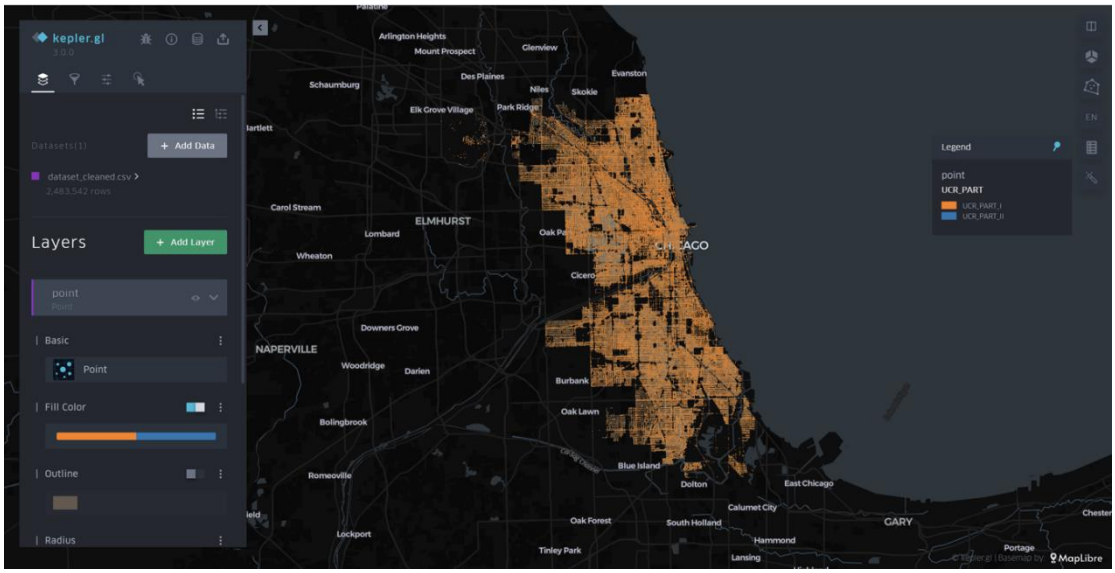


Fig. 5 Map for crimes

Feature Selection

To extract the information from the dataset I first calculated the importance of the features in “ExtraTreesClassifier” (Fig. 6). We can see that the features "District" and "Community Area" are much less important than the other features. Considering that there is more than one area division criterion and spatial features in the data (e.g. District, Community Area, Latitude and Longitude), I extracted these features and fitted decision trees on them to assess their significance. The results (Tab. 2) show that while the accuracy of the separate models based on “District” and “Community Area” is reasonably decent, their F1-scores are not as high as the models based on Latitude and Longitude. We also know that “District” and “Community Area” are defined by different latitude and longitude together, so including them all in the model would lead to multicollinearity. So, after considering all, I decided to choose the final features as “Latitude”, “Longitude”, “Year”, “Month”, “Day”, “Weekday”, and “Time”.

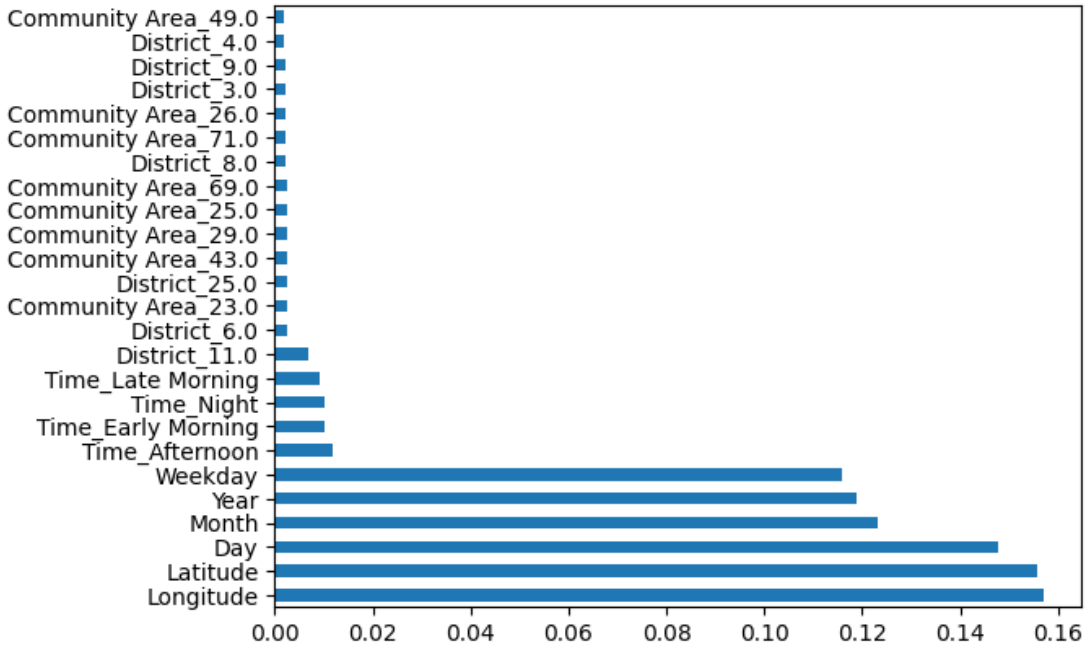


Fig. 6 Importance of the features

Tab. 2 Importance of the features in tree-based models

	Time Data Based	Spatial Data Based		
Features	Year, Month, Weekday, Day, Time	District	Community Area	Latitude, Longitude
Models	Decision Tree	Decision Tree	Decision Tree	Decision Tree
Accuracy	72.8458%	74.0642%	74.0638%	72.8491%
F1-Score	68.8824%	63.0327%	63.0280%	68.8859%

Decision Tree

To implement the decision tree model, I used "SCIKIT LEARN" in python. First, I divided the data into training set and test set which is 70% and 30% respectively. After that, I used the function named "DecisionTreeClassifier" imported from "SCIKIT LEARN" to train the model. After training the model on the training set, it is used to predict the test set and then compared with the test labels to get the actual results. I have chosen accuracy and F1-score as the criteria for model prediction performance evaluation. Here, each leaf node of the decision tree is categorized as a UCR part, i.e. (Part I or Part II). With the visual drawing of the decision tree (Fig. 7) we can see how the tree splits at the first four levels. The accuracy of the decision tree is 64.5720%, with the F1-score equal to 64.8997%. The ROC curve is shown by Figure 8, with AUC = 0.55.

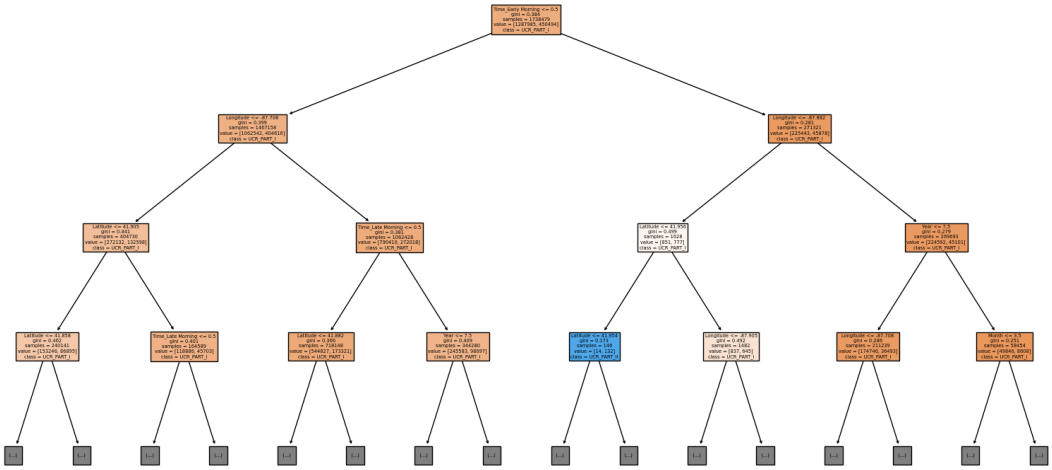


Fig 7. Visualization of decision tree

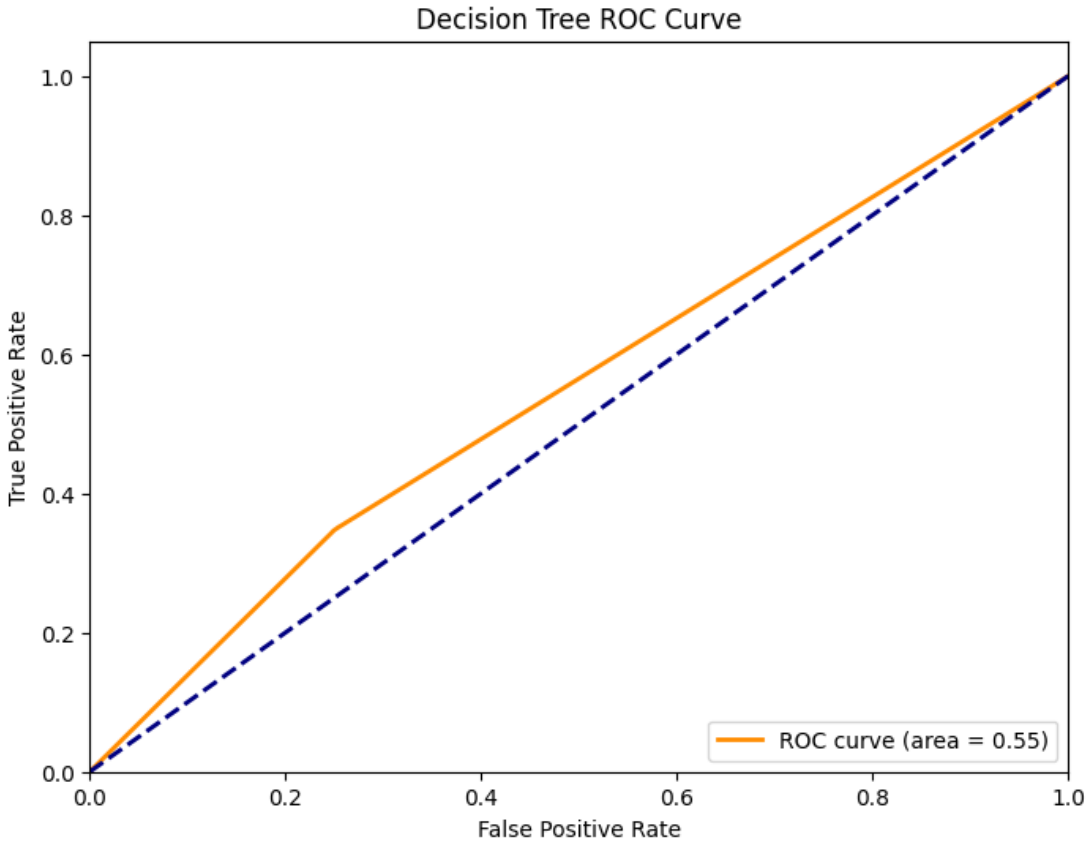


Fig. 8 ROC curve of the decision tree

Random Forest

To implement the random forest model, I used "SCIKIT LEARN" in python. I used training and test sets that have been divided before fitting the decision tree. The reason for controlling the training and test sets unchanged is to allow for more efficient inter-model comparisons. After that, I used a function named "RandomForestClassifier" imported from "sklearn.ensemble" to train the model. Finally, I tuned the hyperparameters by plotting the learning curve (Fig. 9). The optimal parameters are shown in Tab. 3, where “n_estimators” is equal to 181, “max_depth” is equal to 19, “min_samples_split” is equal to 14, “min_samples_split” is equal to 4,“max_features” is equal to 7 and gini is selected for “criterion”. The accuracy of the random forest after using the above hyperparameters can be as high as 75.2330%, with the F1-score equal to 67.3153%. The ROC curve is shown by Figure 10, with AUC = 0.65.

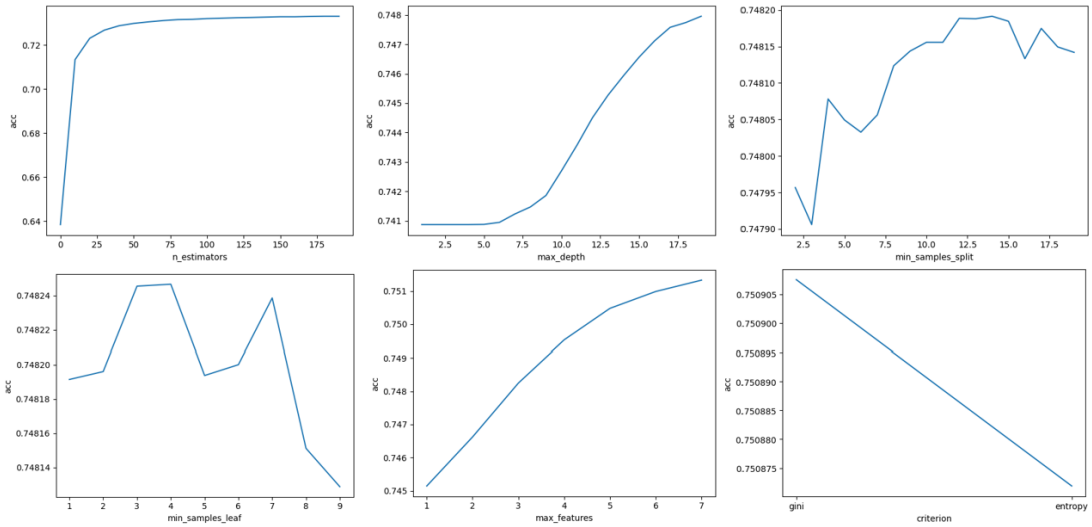


Fig. 9 Learning curves for tuning hyperparameters in "RandomForestClassifier"

Tab. 3 Optimal values for each hyperparameter in "RandomForestClassifier"

Parameter	Value
n_estimators	181
max_depth	19
min_samples_split	14
min_samples_leaf	4
max_features	7
criterion	gini

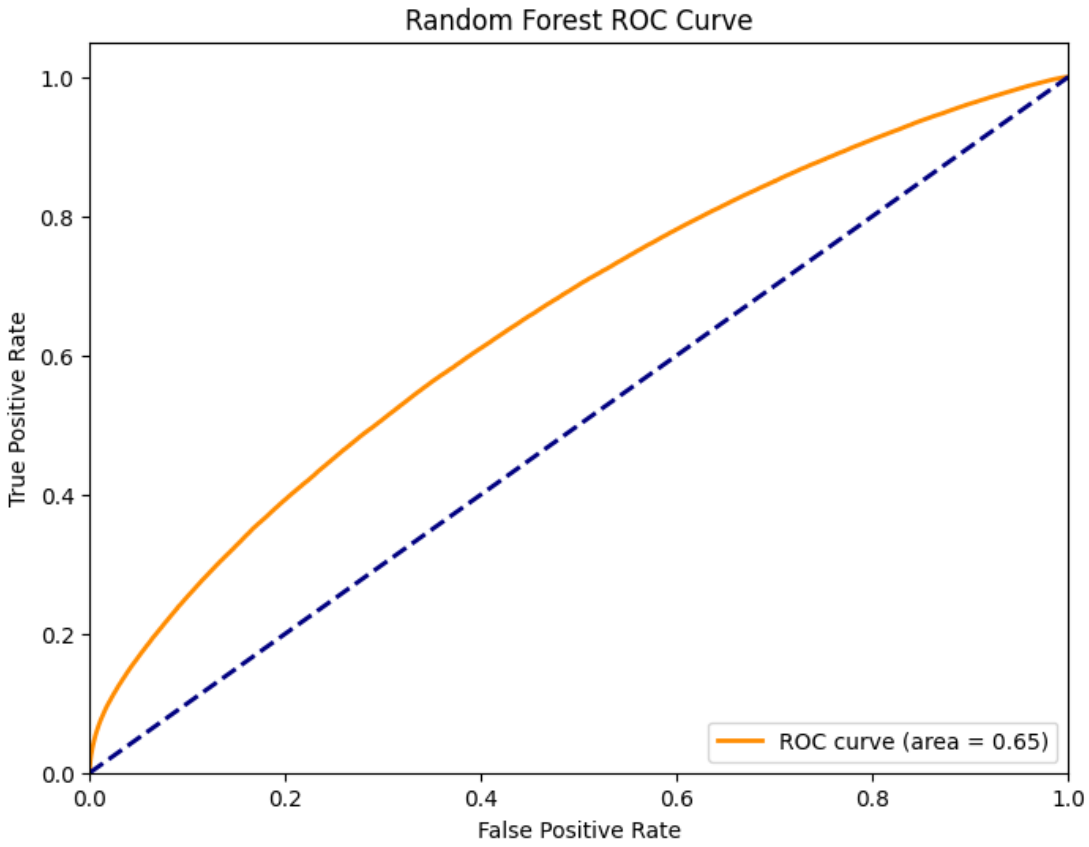


Fig. 10 ROC curve of the random forest

K-Nearest Neighbor

To implement the random forest model, I used "SCIKIT LEARN" in python. I still used training and test sets that have been divided before fitting the decision tree. Then I used the "KNeighborsClassifier" function provided by "sklearn.neighbors" to train the model. KNN has many different parameters like “n_neighbors”, “metric”, “metric_params”, “weights”, “algorithm”, “leaf_size”, “p”, and “n_jobs”. I focused on “n_neighbors” and used the Elbow method to find the optimal value (Fig. 11). The parameter “n_neighbor” = 20 provided us with an optimal accuracy of 73.6901%, with the F1-score equal to 65.0460%. The ROC curve is shown by Figure 12, with AUC = 0.58.

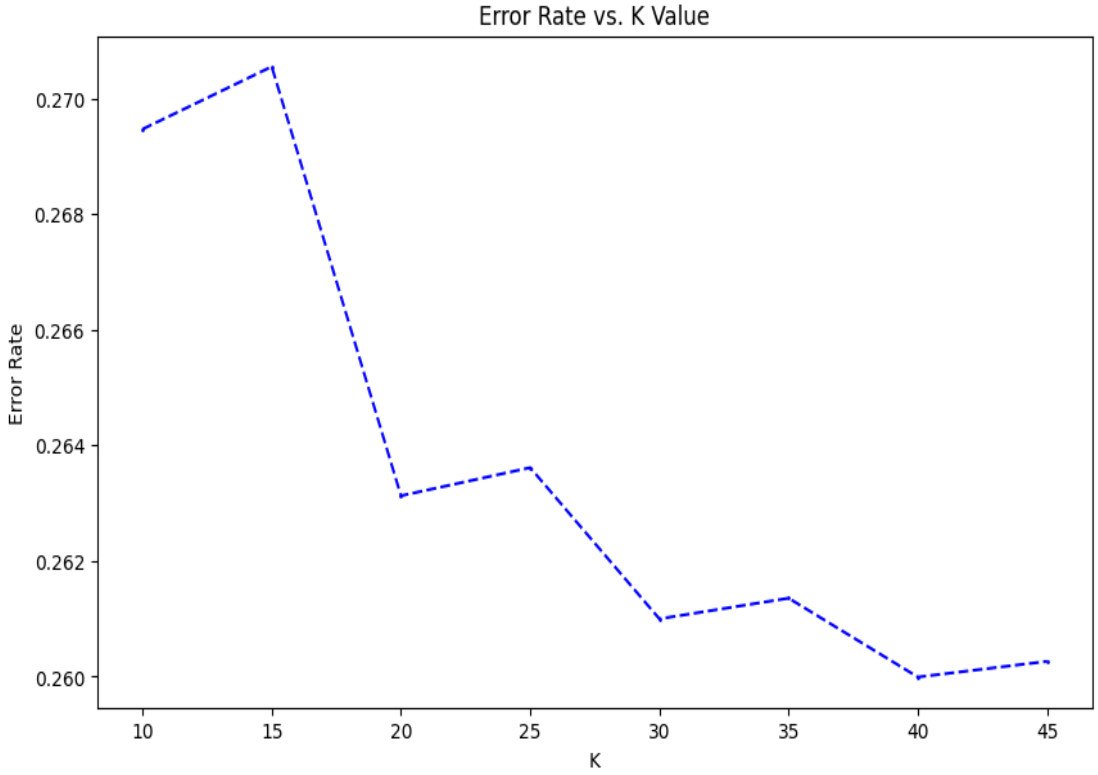


Fig. 11 Elbow method to find the optimal value of "n_neighbor"

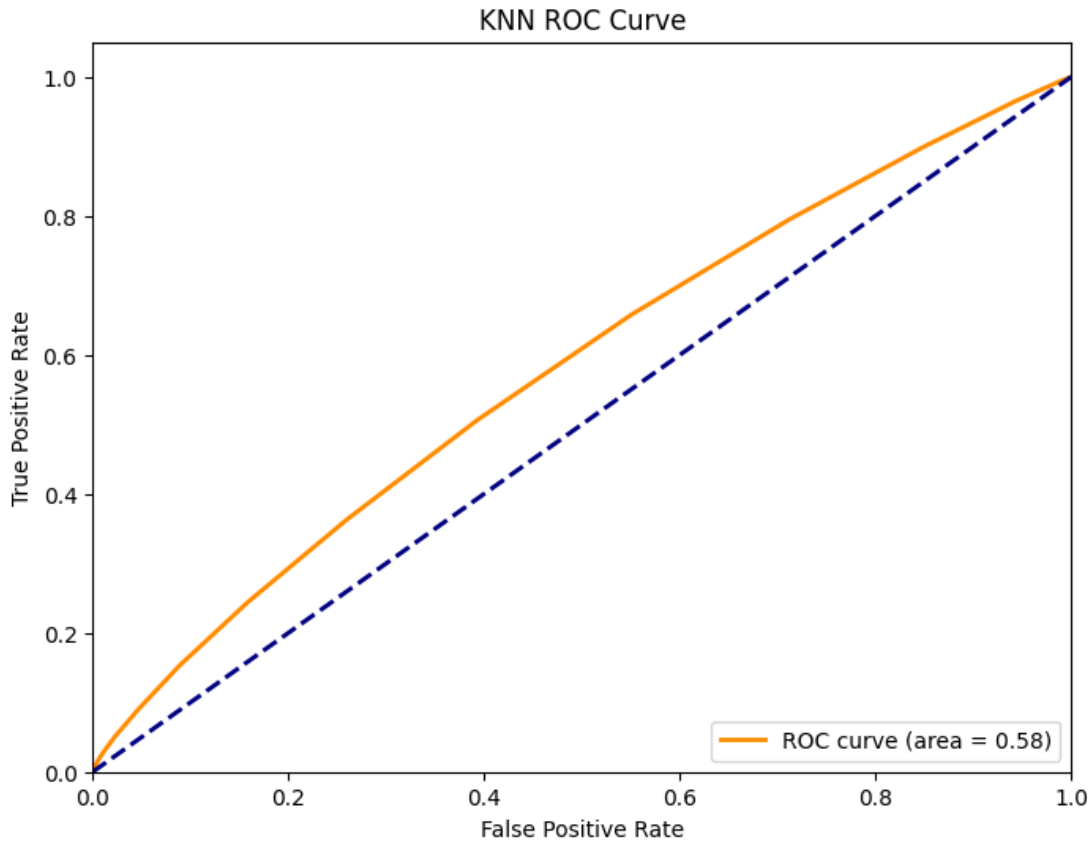


Fig. 12 ROC curve of KNN

Oversampling Dataset

The oversampling method enhances the balance of the dataset by creating new samples from the minority class. There are numerous functions available for balancing an imbalanced class using the oversampling method. In this project, I employed the SMOTE oversampling technique and random oversampling. I used the "SMOTE" and "RandomOverSampler" functions provided by "imblearn.over_sampling" to oversample the unbalanced data respectively.

Undersampling Dataset

Undersampling is applied to the majority classes in an imbalanced dataset. It undersampled the majority classes by compensating minority classes. This method instructs machine learning models to avoid bias and not to overlook false positives. I used the "RandomOverSampler" functions provided by "imblearn.over_sampling" to randomly undersample the unbalanced data.

The results (Tab. 4) show that the original best performing classifier, random forest, did not give better results after the introduction of different oversampling and undersampling techniques.

Tab. 4 Result of different sampling techniques

	Without Sampling	Under Sampling	Over Sampling	
Algorithm		Random	Random	SMOTE
Accuracy	73.3832%	60.5523%	64.8428%	62.6232%
F1-Score	68.1152%	62.9203%	66.3592%	64.5234%

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used dimensionality reduction method for machine learning, which achieves dimensionality reduction by finding the main directions of variation in the data (i.e., the directions of maximum variance of the data) and projecting the data onto these directions. PCA not only effectively reduces the complexity of the data, but also reveals the intrinsic relationship between a set of variables. This makes PCA an important tool in exploratory data analysis and predictive modelling. Therefore, I performed PCA analysis on the raw data by using the "PCA" function provided by "sklearn.decomposition" without using the information of whether the crimes are UCR Part I or Part II. In addition, I plotted a scatter plot of the first score versus the second score. Different colours were used to indicate UCR Part I and UCR Part II. According to Figure 13, we can see that the scatter plots of the data after dimensionality reduction by PCA on the first and the second principal components are not clearly distinguishable. This somewhat foretells that the models constructed based on these two principal components may not achieve the desired performance.

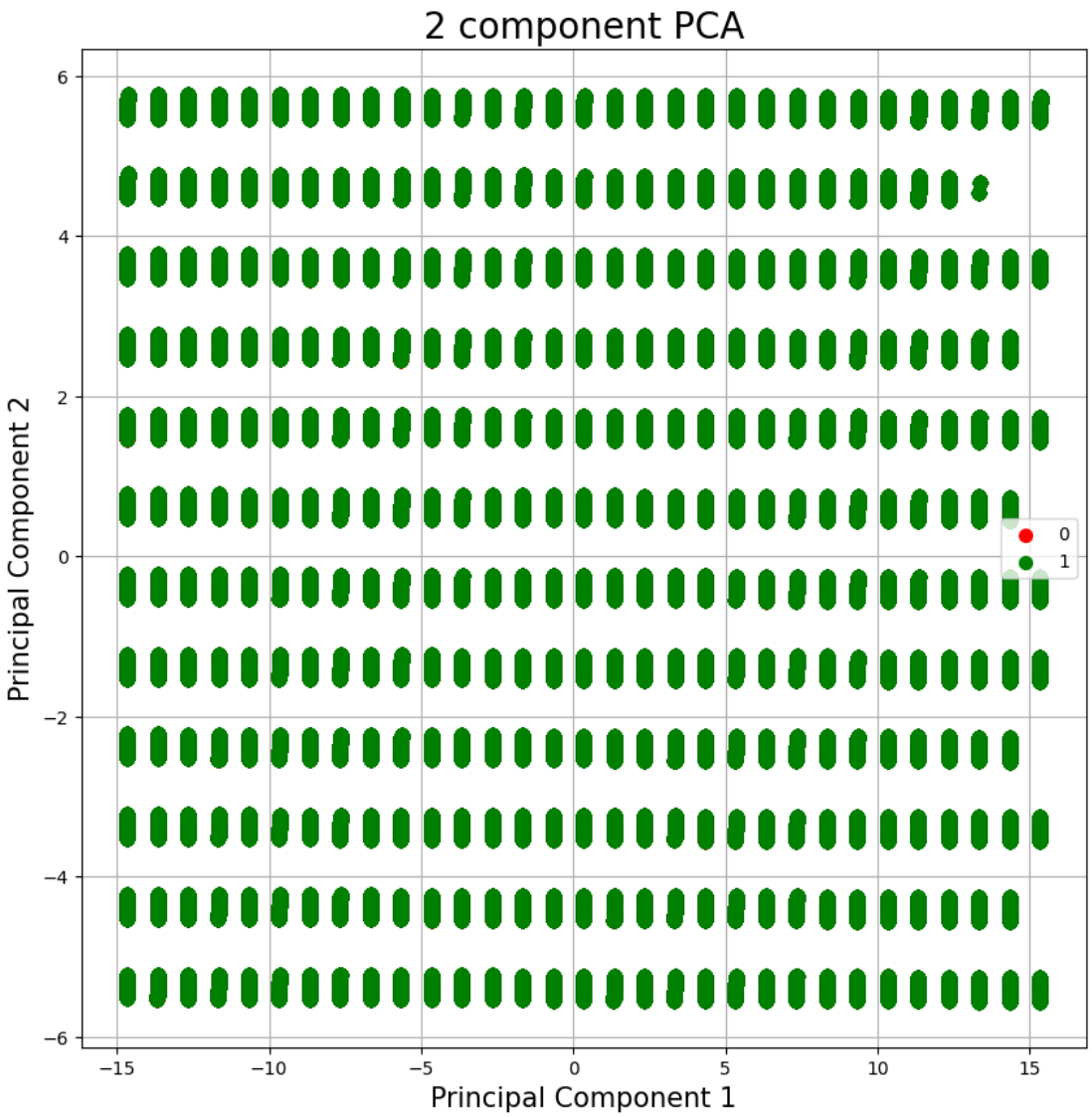


Fig. 13 Scatter plot of principal component 1 vs. principal component 2

Decision Tree with PCA

As mentioned before, I used the function named "DecisionTreeClassifier" imported from "SCIKIT LEARN" again in this section to fit the decision tree based on the first and second principal components. The accuracy of this decision tree is 64.0400%, with the F1-score equal to 63.9095%.

Random Forest with PCA

I employed the “RandomForestClassifier” function from the “sklearn.ensemble” library once more in this section to train a random forest model using the first and second principal components. The accuracy of this random forest model is 67.0243%, with the F1-score equal to 65.5125%. By looking at Table 5, it can be seen that there is no improvement in the predictive performance of both decision trees and random forests fitted on the basis of the first and second principal components.

Tab. 5 Model performance with or without PCA

	Without PCA		With PCA	
Algorithm	Decision Tree	Random Forest	Decision Tree	Random Forest
Accuracy	64.5720%	73.3832 %	64.0400%	67.0243%
F1-Score	64.8997%	68.1152 %	63.9095%	65.5125%

Application – Predictions for 2024

In this section I predicted some types of crimes in Chicago in 2024 based on the model with the best classification performance: the random forest model after tuning the hyperparameters. I have searched to learn about some international events that will be held in Chicago, as shown in Table 6. I thought it is meaningful to use the model to predict the types of crimes for these future events. By predicting, the government and police will know whether the specific type of crime will be UCR Part I or Part II (UCR Part I is more serious). This will help the local police to take precautions in advance and effectively reduce the risk of crime to people.

Because in this section I needed to predict the type of crime in the year 2024, the "Year" feature was removed from the model. I re-fitted the random forest model and used the previously mentioned method of plotting hyperparameter learning curves (Figure 14) to optimise the hyperparameters.

I used the approach mentioned in the data processing section to expand the dataset and achieve prediction of crime types throughout the day by dividing the 24 hours of the day into early morning, late morning, afternoon and night. According to the results (Table 7), regardless of the time period, the type of crime for these events is UCR Part I. This suggests that during the full day that these events are held in these areas, the police need to be well prepared to deal with the possibility of UCR Part I type of crime, which is relatively more serious.

Tab. 6 International events to be held in Chicago in 2024

Activity Name	Day	Location
Chicago Auto Show 2024	2024/2/10	McCormick Exhibition Centre
Expo Chicago 2024	2024/4/11	United States Navy Pier, Illinois
2024 Chicago International Music Competition Final	2024/7/17	Ganz Hall

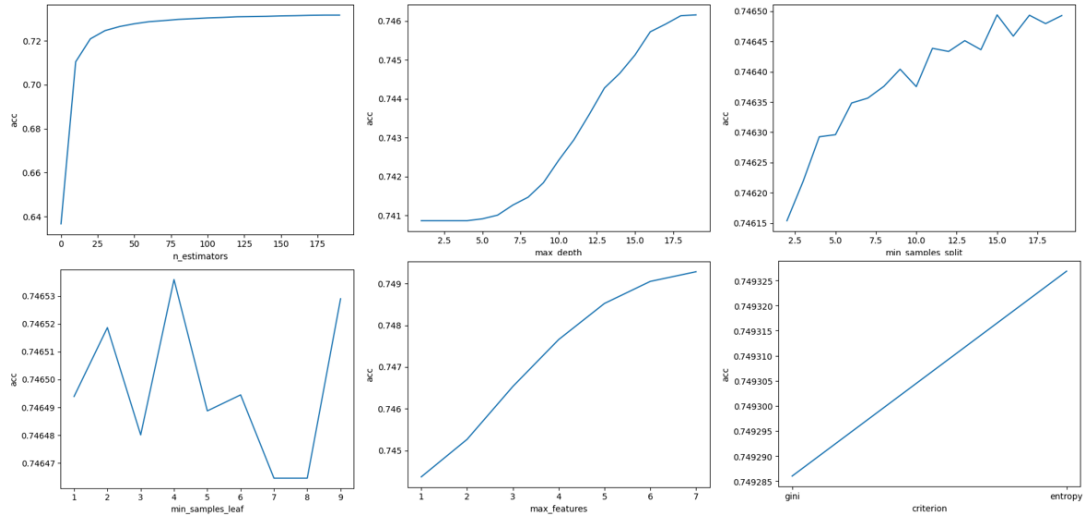


Fig. 14 Learning curves for tuning hyperparameters in "RandomForestClassifier"

Tab. 7 Predicted results

Activity Name	Latitude	Longitude	Month	Day	Weekday	Time	UCR_PART
Chicago Auto Show 2024	41.88323	-87.6324	2	10	Saturday	Early Morning	UCR_PART_I
Chicago Auto Show 2024	41.88323	-87.6324	2	10	Saturday	Late Morning	UCR_PART_I
Chicago Auto Show 2024	41.88323	-87.6324	2	10	Saturday	Afternoon	UCR_PART_I
Chicago Auto Show 2024	41.88323	-87.6324	2	10	Saturday	Night	UCR_PART_I
Expo Chicago 2024	41.8914	-87.5997	4	11	Thursday	Early Morning	UCR_PART_I
Expo Chicago 2024	41.8914	-87.5997	4	11	Thursday	Late Morning	UCR_PART_I
Expo Chicago 2024	41.8914	-87.5997	4	11	Thursday	Afternoon	UCR_PART_I
Expo Chicago 2024	41.8914	-87.5997	4	11	Thursday	Night	UCR_PART_I
2024 Chicago International Music Competition Final	41.8762	-87.6254	7	17	Wednesday	Early Morning	UCR_PART_I
2024 Chicago International Music Competition Final	41.8762	-87.6254	7	17	Wednesday	Late Morning	UCR_PART_I
2024 Chicago International Music Competition Final	41.8762	-87.6254	7	17	Wednesday	Afternoon	UCR_PART_I
2024 Chicago International Music Competition Final	41.8762	-87.6254	7	17	Wednesday	Night	UCR_PART_I

Conclusion

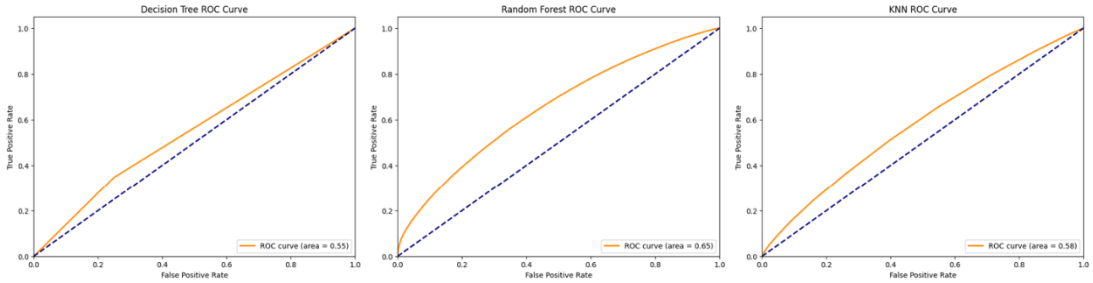
The data in Table 8 shows that random forest has the highest accuracy and F1-score, furthermore, Figure 15 shows that random forest has the largest area under the ROC curve. Therefore, it is reasonable to conclude that random forest is the best model. In this project, I experimented with different features in order to get better predictions such as using weekday, month, and year to predict crime type based on time and using additional features such as latitude and longitude based on location. The results became better, however, not significant enough.

The original dataset was highly imbalanced. UCR Part I has a percentage of 74.1%, which means that if a classifier arbitrarily classifies a crime type into Part I then its accuracy will achieve nearly 74.1%. Therefore, it is more meaningful to compare the F1-scores and the ROC curves between models. Although I tried a number of oversampling and undersampling strategies, none of them brought any improvement in the predictive performance of the model. In addition, using PCA and re-fitting decision trees and random forests did not result in an improvement in prediction performance.

Tab. 8 Performance of different models

Models	Decision Tree	Random Forest	KNN
Accuracy	64.5720%	75.2330%	73.6901%
F1-Score	64.8997%	67.3153%	65.0460%

Fig. 15 ROC curves of different models



Limitations

As can be seen by the map (Fig. 5), not all crimes had a good correlation with parameters such as latitude and longitude. In addition, predicting crime patterns has complicated factors, some of them are related to sociology, economics, even history, and geography. The dataset needs to be expanded.

Future work

In order to enhance the accuracy of prediction, it is crucial to incorporate a broader set of related features into the dataset. These could include economic indicators, demographic

statistics, and weather data, among others. Furthermore, the application of certain deep learning algorithms could potentially improve accuracy. This would involve the careful selection of an appropriate number of layers and filters, along with the optimization of hyperparameters. These combined strategies could provide a more comprehensive and accurate approach to crime prediction.

Reference

Sharma, A., & Singh, D. (2021). Machine learning based analytical approach for geographical analysis and prediction of Boston City crime using geospatial dataset. Geojournal. https://doi.org/10.1007/s10708-021-10485-4

Hossain, S., Abtahee, A., Kashem, I., Hoque, M. M., & Sarker, I. H. (2020). Crime Prediction Using Spatio-Temporal Data. In Communications in Computer and Information Science (Vol. 1235). Springer. https://doi.org/10.1007/978-981-15-6648-6_221

S. Yadav, M. Timbadia, A. Yadav, R. Vishwakarma and N. Yadav, "Crime pattern detection, analysis & prediction," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2017, pp. 225-230.