International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 13 Issue 7, July-2024, Impact Factor: 8.375

Identification of Crime Hotspots Using Machine Learning to Optimize Policing in Crime Prone Cities

Dr. Somnath Roy[1], Priyanki Dutta[2]

ABSTRACT

Criminal activity has long impeded human progress, posing a significant challenge to sustainable living. This issue is exacerbated when Law enforcement resources are limited. In countries like India, where the police-to-population ratio falls short of the United Nations' recommended standard, optimizing the deployment of existing police forces is crucial for crime prevention. This paper presents a solution aimed at enhancing the allocation of police resources across various districts in a city. Our approach determines the likelihood of crime in each district for specific hours, days, and months. Utilizing the Chicago Crime Dataset, we framed the issue as an Imbalanced Classification Problem. We employed several supervised machine learning algorithms, including Logistic Regression, Naive Bayes, K-Nearest Neighbors, Support Vector Machines, Decision Trees, Random Forests, and Gradient Boosting Trees, to evaluate their performance. Among these, the Gradient Boosting Tree algorithm demonstrated superior performance.

INTRODUCTION

As human society progresses towards more sophisticated methods of improving life standards, it remains unable to fully protect its community from the detrimental effects of criminal activities. Crime ranges from minor offenses like roadside theft to serious crimes like homicide, often stemming from complex causes such as a troubled childhood, the influence of delinquent peers, poverty, unemployment, and drug abuse.

The World Health Organization emphasizes, "Crime and violence hinder social and economic development and increase inequalities within and between societies"[1].This highlights the multifaceted impact of crime, not just on individual victims, but on the broader social fabric. Addressing the root causes of crime and fostering a culture of peace and justice are essential steps in creating a sustainable and equitable society for all.

The research presented in [2]demonstrates that an increase in police force correlates with a decline in crime rates. However, in countries like India, where the Police-to-Population ratio falls significantly below the United Nations' standard[3] , it becomes crucial to find methods for effectively allocating police resources across districts based on their susceptibility to crime. Identifying and focusing on regions with higher crime vulnerability, known as "Crime Hotspots,"[4]can help optimize the distribution of the police force to enhance community safety.

The primary objective of this research is to leverage Data Science to promote social welfare by devising a methodology to mitigate the impact of crime. This involves establishing a system that assists the police force in strategically allocating their resources based on the probabilistic crime susceptibility of different regions at specific times. By predicting crime likelihood on a particular day, in a particular month, and at a specific hour, we aim to optimize resource distribution. We employ a variant of the OSEMN (Obtain, Scrub, Explore, Model, Interpret)[5]framework to conduct rigorous data analysis and develop robust computational models. The entire codebase for this study is implemented in Python, utilizing libraries such as Pandas, Seaborn, and Scikit-learn to achieve precise results.

Section	Content Overview
Section II	Related Works in the related field.
Section III	Defines the problem under consideration.

Section IV	Details the methodology employed.
Section V	Presents the findings and results.
Section VI	Concludes the study, highlighting its relevance and potential impact within the Indian context.

RELATED WORK

The application of computer science in identifying patterns in crime has been extensively explored, resulting in significant research contributions. Some studies, like that by Ashokan et al.,[6]focus on specific crime types, while others address crime as a broader category. Data sets such as the Chicago crime data set, utilized in our work, are well-maintained and regularly updated, providing valuable resources for researchers.

For instance, Sathyadevan et al.[7] aggregated crime data from various sources, including the web, social media, and news sites, to create an unstructured dataset analyzed using MongoDB. They employed supervised and apriori algorithms to predict high-crime areas. Similarly, Almanie et al.[8]conducted exploratory data analysis on crime datasets from Denver and Los Angeles, using apriori algorithms to forecast future crime locations. Das et al.[9]used geo-statistical spatial clustering to identify criminal hotspots in India and forecast future crime hotspots based on historical data. Yu et al.[10]focused on burglary hotspots, applying machine learning and deep learning techniques to achieve high accuracy in crime prediction. Shekhar et al.[11]performed a spatio-temporal analysis of crime in Kolar, India, using geo-spatial mapping to identify criminal activity patterns.

Despite these advancements, many crime data mining studies classify results into binary outcomes, indicating whether a crime will occur or not. Few studies have classified hotspots into multiple categories based on the likelihood of crime occurrence at specific times and locations. Our work aims to address this gap by categorizing hotspots into multiple classes, which will help prioritize areas and optimize resource allocation. This is particularly important in countries like India, where the police force is often understaffed. By doing so, our research intends to support police precincts in managing their limited resources more effectively.

Problem description:
This study aims to analyze crime data and identify crime hotspots—regions characterized by elevated criminal activity—within specific temporal contexts: monthly, daily, and hourly. The primary dataset utilized is the Chicago Crime Dataset, accessible through the City of Chicago Data Portal[12]. This dataset comprehensively captures crime records across Chicago from 2001 to the present day, updated daily. For this research, we focused on data spanning from 2020 to 2024 (records current until 13/06/24).

Our methodology involves several key steps: Data Preprocessing, Compression to reshape the dataset into a more suitable format for analysis, explicit preparation of training and test datasets, and the application of machine learning algorithms for classification. Specifically, we aim to classify entries into crime hotspots based on the historical intensity of crimes occurring in those regions during specific time points. In our context, a time point is represented as a one-dimensional vector comprising month, day, and hour values.

Fig. 1. OSEMN variant methodology

Our proposed algorithms utilize these time points to classify each district in the dataset into areas with low, medium, or high crime rates for each specific time point, as illustrated in Fig. 1 of the OSEMN variant methodology.

$$t = [\text{month day hour}] \quad (1)$$

The core challenge is a supervised multi-class classification problem, which inherently faces issues of class imbalance[13]. In the upcoming section, we will delve into the implementation details and the theoretical foundations of our approach.

METHODOLOGY

This study adopts a data science methodology inspired by the OSEMN framework in [14], as detailed in, to ensure comprehensive analysis and optimal results from the data set. OSEMN, a widely recognized data science process, entails sequential stages of Obtain, Scrub, Explore, Model, and Interpret. Our methodology emphasizes the comprehensive OSEMN framework, seamlessly integrating the phases of Obtain, Scrub, Explore, and Model, depicted in Fig. 1, to systematically derive insights and construct robust analytical models.

A. Obtain Data

The original training dataset for our project was sourced from the City of Chicago Data Portal and comprises 10,24,688 entries across 22 features. Key features include Date, Block, Primary Type, Location, Arrest, District, and Year. Among these features, 2 are boolean, 6 are float, 4 are integer, and 10 are string data types. This dataset encompasses all reported crimes in Chicago from 2020 to 2024 (up to June).

Additionally, data from 2017 to 2019 were utilized to create one of the testing samples for our analysis.

B. Scrub Data

This portion has been subdivided for clarity and ease of comprehension.

a) **Handling Missing Values**:

Dealing with missing values is crucial as they can significantly impact model performance. Fortunately, the information set is moderately refined with few lost values. The lost information numbered for a little division of the add up to information set.

Entries containing missing information were therefore removed, resulting in retention of 98.06 % of the complete data.

b) Feature Engineering:

In this phase, we first transformed the "Date" feature into a format suitable for Python code. Subsequently, we performed feature engineering [15] to extract distinct features: "Month", "Day", and "Hour". Originally, the "Date" feature in our dataset encompassed the month, day, and time of each reported crime. To achieve this transformation, we utilized the `apply()` function from the Pandas package[16] and Python's `datetime` module[17].

python:

```
# Transform Date feature into datetime format
data['Date'] = pd.to_datetime(data['Date'])

# Extricate Month, Day, and Hour from Date highlight
data['Month'] = data['Date'].dt.month
data['Day'] = data['Date'].dt.day
data['Hour'] = data['Date'].dt.hour
```

This approach allowed us to efficiently parse and extract temporal components from the original "Date" feature, facilitating further analysis and modeling.

c) Compressing the Dataset:

Next, we proceeded to compress and reshape the dataset into a new format, referred to as `crimeDat` for the remainder of this study. This final dataset, `crimeDat`, was used extensively in our analysis. The pseudocode below outlines the process for creating `crimeDat` from our original dataset `data`.

python:

```
# Pseudocode for creating crimeDat
cols = ['Month', 'Day', 'District', 'Hour']
crimeDat = data.groupby(cols)  # Group data by Month, Day, District, and Hour
crimeDat = crimeDat.aggregateByCrimeCount()  # Aggregate by counting crimes
crimeDat = crimeDat.descSort(by='District')  # Sort by District in descending order
```

After compression and reshaping, `crimeDat` consists of 44,385 entries and includes 5 features: Month, Day, District, Hour, and CrimeCount. Table I provides the first five entries of `crimeDat`. This streamlined dataset allowed us to focus our analysis on essential crime patterns and correlations within specific temporal and spatial contexts.

C. Explore

In this phase, we engineered a new target feature, `crimeDat['Alarm']`, for the `crimeDat` dataset. This feature categorizes entries into 0 (Low alarm), 1 (Medium alarm), or 2 (High alarm) based on their alarm rate.

Labeling entries with these alarm levels involves a subjective approach that can vary among different methodologies. In our approach, we at first plotted a histogram (Fig 2) to outline the dispersion of values in 'crimeDat ['CrimeCount']'.

This distribution closely resembled a normal distribution, which guided our intuitive classification of entries within `crimeDat`.

Table I : initial 5 rows of crimeDat

Month	Day	Hour	District	Crime_Count
6	0	21	31.0	1
8	4	9	31.0	1
1	5	10	31.0	1
6	5	18	31.0	1
8	2	7	25.0	10

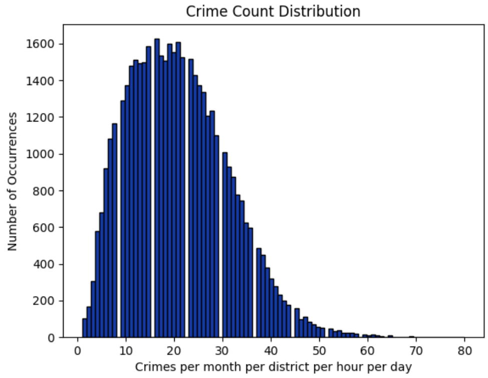


Fig. 2. Distribution of Crimes

This intuitive classification process enabled us to categorize and analyze crime rates effectively within our dataset. We classified all entries in `crimeDat` based on their values in `crimeDat['CrimeCount']` relative to the distribution's mean and standard deviation. Entries falling within ±0.75 standard deviations of the mean were labeled as 1 (Medium alarm). Entries below this range were classified as 0 (Low alarm), and those above were labeled as 2 (High alarm). The pseudocode below outlines how these target labels were determined:

python:

```
# Pseudocode for setting target names in crimeDat['Alarm']
mean = crimeDat['CrimeCount'].mean()
std_dev = crimeDat['CrimeCount'].std()
threshold_low = mean - 0.75 * std_dev
threshold_high = mean + 0.75 * std_dev

for entry in crimeDat:
    if entry['CrimeCount'] < threshold_low:
        entry['Alarm'] = 0  # Low alarm
    elif entry['CrimeCount'] > threshold_high:
        entry['Alarm'] = 2  # High alarm
    else:
```

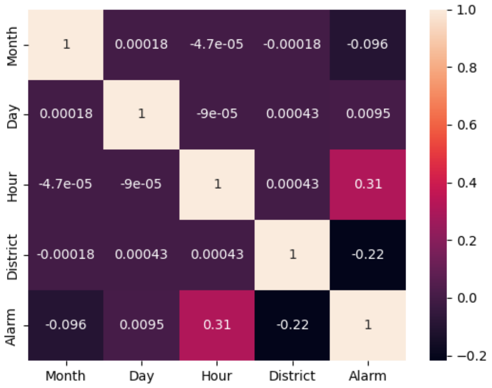
entry['Alarm'] = 1 # Medium alarm

This approach allowed us to classify entries in `crimeDat` into distinct alarm levels based on their crime count values, facilitating further analysis and model training.

Fig. 3 depicts the correlation matrix [18] among the features in `crimeDat`. From the results, we can deduce:

- There is negligible correlation observed between most features. This is beneficial because high multi-collinearity can adversely affect certain machine learning algorithms.

- The feature "Hour" shows the highest positive correlation with the target feature "Alarm."



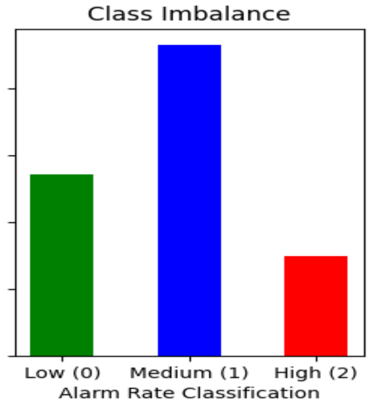


Fig. 3. Correlation Matrix   Fig. 4. Class Imbalance in crime

Fig. 4 outlines the lesson awkwardness show in our classification issue. Class imbalance arises when the sample sizes for each class type differ significantly.In our dataset:

- The number of samples classified as "medium" (`crimeDat['Alarm'] = 1`) is substantially higher compared to those classified as "low" (`crimeDat['Alarm'] = 0`) and "high" (`crimeDat['Alarm'] = 2`).

- Class imbalance poses challenges for accurate classification models and often leads to the "Accuracy Paradox,"[19]where high overall accuracy may mask poor performance on minority classes.

These visualizations provide critical insights into the dataset's characteristics and challenges, guiding the selection and evaluation of appropriate machine learning algorithms for further analysis.

D. Model

Having prepared the crimeDat dataset for classification, the next step is to utilize this dataset to perform our classifications.

In the modeling step, we begin by establishing a mathematical proof to affirm that "The degree of criminal presence at a given time point (month, day, and hour) is dependent on the area or locale in address." The verification is organized as takes after:

Let $t_i$be a time point vector as said in equation no 1. By vertically stacking these vectors, we construct the "Time Space Matrix" T:

T= t0+t1+…+tn(2)
Here, the symbol "+" denotes vertical concatenation.
Referring to Table I, we define "crimeDat" as:
crimeDat=T+D+A   (3)
where:

- A represents the "Alarm" vector, our target feature.
- D represents the "District" vector, encompassing the 31 districts in Chicago.

From equation (3), we derive equation (4):
$t_i+d\in$ crimeDat (4)
where $d\in D$. Equation (4) signifies that the combination of a "Time point vector" t_i and a district d encapsulates the independent features of crimeDat.
Another, we apply a machine learning calculation algo to our information set. For a given possible value a of crimeDat[Alarm]:

$t_i+d_{algo}=a$ (5)

Keeping algo and t_i constant, we infer that a (Alarm Rate) is dependent on d or the district in question. This implies that the Alarm Rate at a given time point t_i depends on the specific district d.

Consequently, not all districts require the same level of police force for patrolling at any given time point. The subsequent step involves employing machine learning algorithms to effectively classify[20]each district based on its vulnerability to crime at specific time points.

The scope of our work encompasses various machine learning algorithms, including Gradient Boosting Tree (GBT) [21] , Random Forest (RF)[22], Decision Tree (DT)[23] , K Nearest Neighbor (knn)[24], Support Vector Machine (SVM)[25], Logistic Regression (LR)[26], and Naive Bayes (NB)[27]. It was observed that tree-based algorithms provided significant advantages over non-tree-based models due to the imbalanced nature of the dataset.

EXPERIMENTAL RESULT

In machine learning issues, assessing demonstrate execution is crucial. According to the "No Free Lunch"[28]theorem , comparing multiple machine learning models across various parameters and evaluation metrics is an essential process. In our supervised classification problem, we utilized several metrics to assess model performance, including accuracy, precision, recall, F1 score, and unweighted average recall (UAR)[29].

Equation (6): Accuracy
Accuracy=(TP+TN)/TP+FP+TN+FN

- **TP (True Positives):** Number of positive tests accurately recognized as positive.
- **TN (True Negatives):** Number of negative samples correctly identified as negative.
- **FP (False Positives):** Number of negative samples incorrectly identified as positive.
- **FN (False Negatives):** Number of positive samples incorrectly identified as negative.

Equation (7): Precision
Precision=TP/(TP+FP)

- **Precision:** Describes the likelihood of a sample belonging to class X if it has been predicted as belonging to class X. It centers on rightness of positive forecasts.

Equation (8): Recall (or Sensitivity)
Recall=TP/(TP+FN)

- **Recall:** Describes the likelihood of the model correctly classifying a sample as belonging to class X, given that the sample actually belongs to class X. It centers on completeness of positive forecasts.

Equation (9): F1 Score
F1 Score=(2×Precision×Recall)/Precision+Recall
F1 Score: Harmonic mean of precision and recall. It gives a single metric that equalizations both exactness and review, emphasizing the execution on positive tests.

- .

Equation (10): Unweighted Average Recall (UAR)
UAR=(R1+R2+R3)/3

UAR: Average of the recall values across all classes being classified. It provides an overall measure of how well the model performs in identifying instances of each class, without giving more weight to any specific class.
Ri: Recall value for the i^{th} class, where i can represent different classes being classified.

These metrics are crucial in checking the performance of a administrative classification model. They help in understanding how well the model distinguishes between classes, balances precision and recall, and handles class imbalance in the dataset. Each metric provides unique insights into different aspects of the model's effectiveness and can guide improvements in model training and evaluation.

The models were trained on 75% of the crimeDat dataset and subsequently tested on three distinct samples of data:

Sample 1: This sample consists of 25% of crimeDat, maintaining the original class imbalance observed in the dataset. Class imbalance refers to the uneven distribution of instances across different classes (e.g., low, medium, high alarm levels).

Sample 2: This sample also includes 25% of crimeDat but is processed to mitigate class imbalance.This is accomplished through the oversampling methods, which point to adjust the information set by expanding the number of occasions in the minority course or classes.

Sample 3: This sample comprises all crime records from the years 2017 to 2019. Unlike Samples 1 and 2, which are subsets of crimeDat from the years 2020 to 2024, Sample 3 includes data from earlier years. This allows for testing the model's performance on a broader range of historical data.

The results were obtained using k=5 for K-Nearest Neighbors (KNN), "multinomial" logistic regression for LR, a "linear" kernel for SVM, Gaussian Naive Bayes (NB), and 1000 trees for both Random Forest (RF) and Gradient Boosting Tree (GBT) algorithms. Table II illustrates that each of the three testing samples yielded different best performing models. Alongside accuracy, Table II showcases F1 scores and UARs for each model. F1 scores were macroaveraged across all models to assess effectiveness on minor classes[30], particularly those classified as "high alarm," which is crucial for ensuring citizen safety.GBT achieved the most noteworthy precision in Test 1, whereas RF exceeded expectations in Test 2. KNN illustrated the most elevated exactness in Test 3.However, due to dataset imbalance, relying solely on accuracy is inconclusive. Hence, macroaveraged F1 scores and UARs were crucial metrics. GBT, RF, and KNN achieved the highest macroaveraged F1 scores across the samples, indicating their ability to recognize and classify minority classes effectively. Higher UAR values reflect a model's proficiency in classifying samples into their respective classes.

Comparing UAR metrics across samples revealed that GBT, RF, and GBT performed best in Samples 1, 2, and 3 respectively. Thus, the top three models identified in this research are GBT, RF, and KNN. SVM performed poorly with a null UAR score, failing to recognize the "high alarm" minority class and making no predictions for it. This failure is critical as models must predict high crime regions to ensure public safety. Another concern highlighted by this research is the misclassification of "high alarm" regions as "low alarm," which could lead to inadequate police deployment. GBT demonstrated exceptional performance with very low misclassification percentages across all samples (0.04%, 0.02%, and 0%). Therefore, selecting a computational model that achieves high accuracy across all three classes, maintains a high macroaveraged F1 score, and minimizes misclassification of "high alarm" regions as "low alarm" regions is essential.

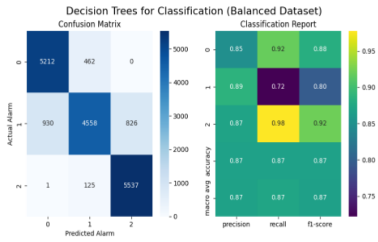
GBT effectively fulfills these criteria better than other models evaluated in this study.

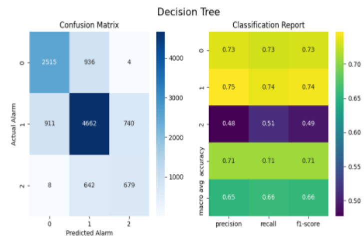
Table II: Comparing Model Performances

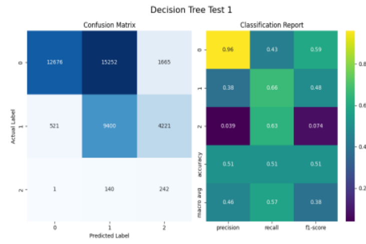
Machine LearningModel	**Performance**								
	Sample1			Sample 2			Sample 3		
	Accuracy	F1 Score	UAR	Accuracy	F1 Score	UAR	Accuracy	F1 Score	UAR
Gradient Boosting Tree	79.32	0.79	0.776	78.65	0.79	0.7896	55.24	0.5665	0.588
Random Forest	77.62	0.77	0.74	89.15	0.89	0.8962	51.06	0.56	0.574
Decision Tree	70.79	0.71	0.6522	86.72	0.86	0.8681	50.587	0.55	0.4596
KNN	75.40	0.75	0.714	78.91	0.78	0.789	69.68	0.57	0.56
SVM	56.32	0.42		51.82	0.52	0.51	46.77	0.52	0.417
Naive Bayes	64.10	0.59	0.417	53.82	0.53	0.535	46.67	0.55	0.425
Logistic Regression	63.36	0.58	0.45	52.036	0.52	0.516	46.81	0.55	0.426

Comparing Model Performances Based on HeatMap:

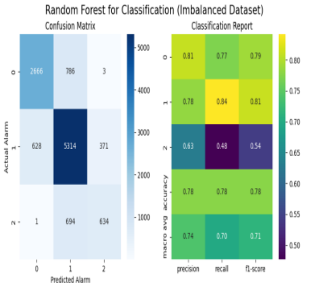
Decision Tree Plot:

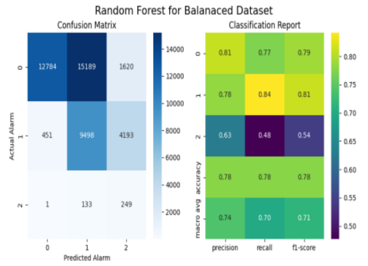


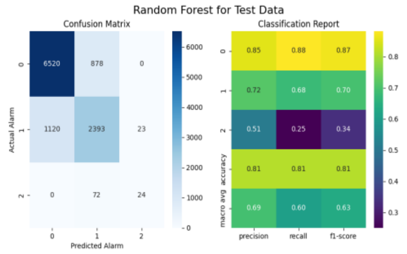




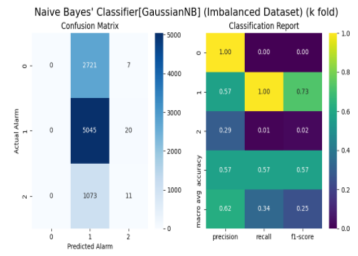
Random Forest Plot:

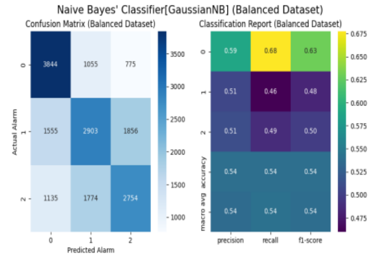


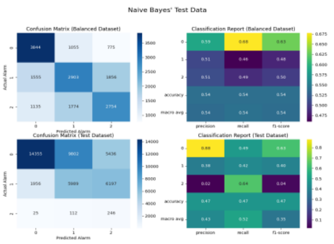




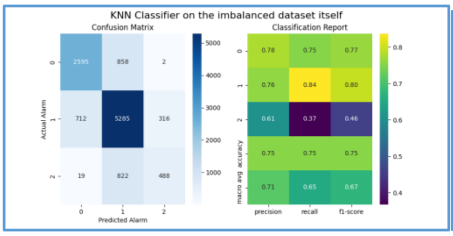
Naive Bayes Plot:

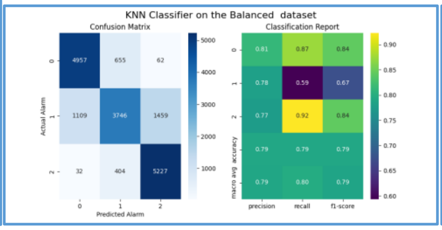


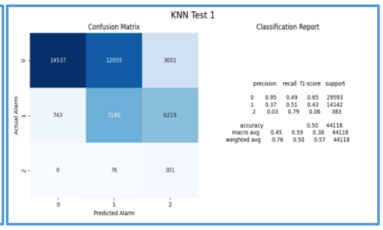




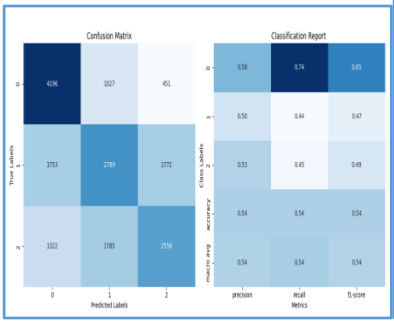
KNN Classifier Plot:

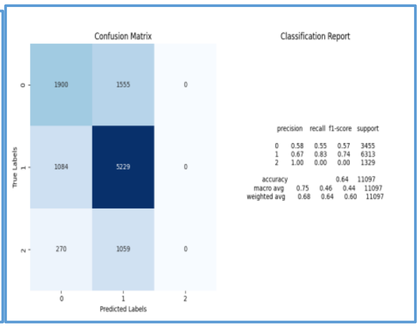


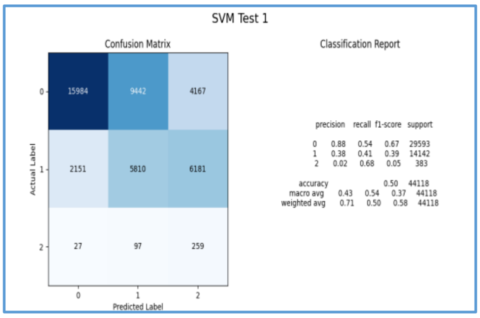




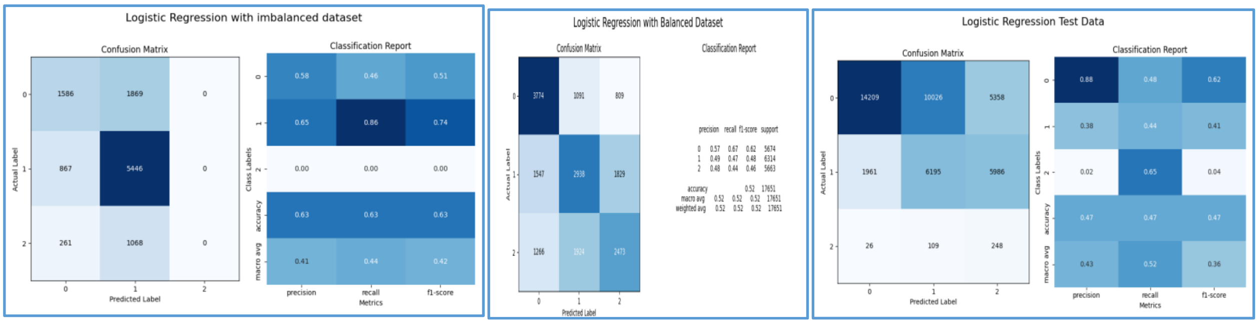
SVM Plot:



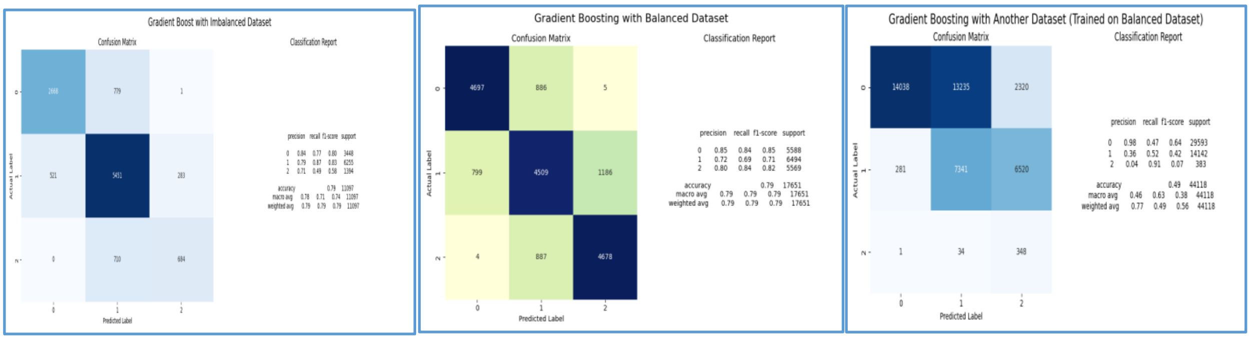




Logistric Regreesion plot:



Gradient Boosting Tree Plot:



CONCLUSION

Preventing criminal activities stands among the foremost global priorities in fostering secure, cohesive communities that shield individuals from both physical and psychological harm. This paper presents a solution to address the pervasive issue of inadequate police presence in areas plagued by frequent criminal incidents. It introduces a novel approach to analyzing crime data, aiming to optimize the deployment of law enforcement resources effectively. The methodology outlined in this study offers a fresh perspective on how crime-related datasets can be leveraged in research, demonstrating a method to strategically allocate police resources based on the temporal dynamics of criminal activity across different districts. The dataset used originates from Chicago and provides a comprehensive record of reported crimes. However, similar publicly accessible crime data from India could potentially facilitate the implementation of this approach within the Indian context. This adaptation would be particularly beneficial for Indian law enforcement, given the apparent shortage of police personnel relative to the country's crime challenges. Furthermore, employing machine learning techniques in this study offers profound insights into crime prevention strategies. By analyzing historical crime patterns, the research highlights opportunities to optimize resource allocation, thereby enhancing the effectiveness of law enforcement efforts. In summary, this paper underscores the importance of data-driven approaches in mitigating crime impacts and advocates for the judicious utilization of existing resources to bolster public safety and security in communities worldwide.

REFERENCES

[1]. “The Links between Violence, Inequality and Productivity “, UNDP and UNU_-WIDER, July 2021.
[2]. M.-J. Lin, "More police, less crime: Evidence from US state data," International Review of Law and Economics, vol. 29, pp. 73–80, June 2009.
[3]. V. P. Sriharsha Devulapalli, "India's police force among the world's weakest," Available at: https://www.livemint.com/news/india/india-s-police-force-among-the-world-s-weakest-1560925355383.html, Accessed: July 19, 2019.
[4]. National Academies of Sciences, Engineering, and Medicine, "Proactive Policing: Effects on Crime and Communities," D. Weisburd and M. K. Majmundar (Eds.), Washington, DC: The National Academies Press, 2018. Available at: https://www.nap.edu/catalog/24928/proactive-policing-effects-on-crime-and-communities.
[5]. J. Brownlee, "How to work through a problem like a data scientist," Available at: [https://machinelearningmastery.com/how-to-work-through-a-problem-like-a-data-

scientist/](https://machinelearningmastery.com/how-to-work-through-a-problem-like-a-data-scientist/), Accessed: May 13, 2019.

[6]. V. Ashokan and O. R. Murthy, "Comparative evaluation of classifiers for abnormal event detection in ATMs," in 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), IEEE, 2017, pp. 1330–1333.

[7]. S. Sathyadevan, S. Gangadharan et al., "Crime analysis and prediction using data mining," in 2014 First International Conference on Networks & Soft Computing (ICNSC2014), IEEE, 2014, pp. 406–412.

[8]. T. Almanie, R. Mirza, and E. Lor, "Crime prediction based on crime types and using spatial and temporal criminal hotspots," arXiv preprint arXiv:1508.02050, 2015.

[9]. S. Das and M. R. Choudhury, "A geo-statistical approach for crime hotspot prediction," International Journal of Criminology and Sociological Theory, vol. 9, no. 1, 2016.

[10]. C.-H. Yu, M. W. Ward, M. Morabito, and W. Ding, "Crime forecasting using data mining techniques," in 2011 IEEE 11th International Conference on Data Mining Workshops, IEEE, 2011, pp. 779–786.

[11]. S. Shekhar et al., "Spatial analysis of crime report datasets," Prof. Shashi Shekhar's Spatial Computing Research Group, 2008.

[12]. City of Chicago, "Crimes - 2001 to present," City of Chicago Data Portal, Available at: https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data, Accessed: May 11, 2019.

[13]. R. Longadge and S. Dongre, "Class imbalance problem in data mining review," arXiv preprint arXiv:1305.1707, 2013.

[14]. K. Dineva and T. Atanasova, "Osemn process for working over data acquired by IoT devices mounted in beehives," Current Trends in Natural Sciences, vol. 7, no. 13, pp. 47–53, 2018.

[15]. Wikipedia contributors, "Feature engineering — Wikipedia, the free encyclopedia," 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Feature_engineering&oldid=909673146, Accessed: July 21, 2019.

[16]. W. McKinney et al., "Data structures for statistical computing in Python," in Proceedings of the 9th Python in Science Conference, vol. 445, Austin, TX, 2010, pp. 51–56.

[17]. Python Software Foundation, "8.1. datetime basic date and time types," Python 3.3.7 Documentation, 2019. [Online]. Available: https://docs.python.org/3.3/library/datetime.html, Accessed: July 22, 2019.

[18]. J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson Correlation Coefficient," Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–4. [Online]. Available: https://doi.org/10.1007/978-3-642-00296-0_5.

[19]. Wikipedia contributors, "Accuracy paradox — Wikipedia, the free encyclopedia," [Online]. Available: https://en.wikipedia.org/w/index.php?title=Accuracy_paradox&oldid=887868751, Accessed: July 10, 2019.

[20]. S. Palaniswamy and S. Tripathi, "Geometrical approach for emotion recognition from facial expressions using 4D videos and analysis on feature-classifier combination," International Journal of Intelligent Engineering and Systems, vol. 10, pp. 30–39, April 2017.

[21]. G. Moisen, "Classification and regression trees," In: Jørgensen, Sven Erik; Fath, Brian D. (Editor-in-Chief), Encyclopedia of Ecology, volume 1, Oxford, UK: Elsevier, pp. 582–588, 2008.

[22]. A. Liaw, M. Wiener et al., "Classification and regression by random forest," R News, vol. 2, no. 3, pp. 18–22, 2002.

[23]. S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660–674, 1991.

[24]. S. Larsen and L. N. Kanal, "Analysis of k-nearest neighbor branch and bound rules," Pattern Recognition Letters, vol. 4, no. 2, pp. 71–77, 1986.

[25]. S. R. Gunn et al., "Support vector machines for classification and regression," ISIS Technical Report, vol. 14, no. 1, pp. 5–16, 1998.

[26]. D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, "Logistic regression," Springer, 2002.

[27]. I. Rish et al., "An empirical study of the naive Bayes classifier," in IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, no. 22, pp. 41–46, 2001.

[28]. D. H. Wolpert, "The Supervised Learning No-Free-Lunch Theorems," London: Springer London, 2002, pp. 25–42. [Online]. Available: https://doi.org/10.1007/978-1-4471-0123-9_3.

[29]. M. Hossin and M. Sulaiman, "A review on evaluation metrics for data classification evaluations," International Journal of Data Mining & Knowledge Management Process, vol. 5, no. 2, p. 1, 2015.

[30]. V. Van Asch, "Macro- and micro-averaged evaluation measures [Basic draft]," Belgium: CLiPS, pp. 1–27, 2013.