**CHICAGO CRIME ANALYSIS- 2022 REPORT**

**GROUP J: TEJASHREE RAJAGOPALAN**

------------------------------------------------------------------------------------------------------------------------------------------**EXECUTIVE SUMMARY**

This report offers a thorough examination of crime statistics from Chicago city in the year 2022. Our goal was to employ sophisticated data analytics methods to uncover trends, pinpoint areas of concern, and offer valuable insights for law enforcement agencies. Through a multifaceted approach involving data processing, exploratory analysis, feature engineering, and using supervised machine learning algorithms like Decision Tree Classifier, Random Forest, Gradient Boosting and Logistic Regression, Our team aimed to support the improvement of public safety measures and the development of more effective crime prevention strategies in the city.

**INTRODUCTION**

The ongoing challenge of crime threatens societal well-being across various fronts, including public safety, community cohesion, and economic stability. In response, our project concentrated on analyzing crime data from Chicago in the year 2022, a city renowned for its complex urban dynamics.

Utilizing data-driven methodologies to achieve the following objectives:

* Identify the top 5 crimes of 2022.
* Analyze monthly frequencies of these crimes to facilitate proactive planning.
* Determine high-risk time periods for strategic resource deployment.
* Implement supervised machine learning techniques to classify crime hotspots.

Our goal is to furnish law enforcement with actionable intelligence, thereby enhancing crime prevention efforts and public safety measures, ultimately cultivating a safer environment for all residents of Chicago.

**METHODOLOGY**

* DATASET DESCRIPTION: The dataset chosen for this project encompasses crime incidents reported exclusively in the year 2022 within the city of Chicago. Sourced from the Chicago Police Department Data Portal, this dataset consists of 239,329 rows and 22 columns. The dataset is provided in .csv format, ensuring ease of accessibility and compatibility for analysis purposes.
* DATA ACQUISITION: In our Jupyter Notebook environment, we imported necessary libraries for various tasks. For data handling, we utilized Pandas and NumPy. Matplotlib and Seaborn were employed for visualization purposes, while Datetime was used for date handling. Scikit-learn was employed for modeling, and we also utilized Joblib for utility functions. After importing these libraries, we loaded the dataset to inspect it and checked its information.

A screenshot of a computer

Description automatically generated A white background with text

Description automatically generated with medium confidence

* DATA PRE-PROCESSING: Examining the Chicago Crime Dataset reveals the presence of missing data, particularly in features related to the geographical location of crime scenes. These missing values stem from the dataset's reliance on initial accounts from involved individuals. Acknowledging that reports provided to the Police Department may lack specific location details and that initial crime classifications are subject to change pending further investigation, the dataset's integrity remains paramount. With 23,043 missing values across various features, traditional statistical methods for imputation are impractical. Consequently, the most suitable approach is to remove these missing values, ensuring data integrity and accuracy for subsequent analysis and modeling.
* EXPLORATORY DATA ANALYSIS: We identified distinct crimes in the Chicago dataset using the primary time feature and populated them accordingly. Afterward, we plotted the top 5 crimes in Chicago.

**Result:** In 2022, the most common top five crimes in Chicago were theft, battery, criminal damage, motor vehicle theft, and assault.

A graph of blue bars

Description automatically generated

* FEATURE ENGINEERING:
* We converted the date/time to datetime objects, extracted month, day, and hour, and created a column for month. Then, we plotted the top 5 crimes based on the

months.

**Result:** The months of July, August, September, and October exhibited the most significant spike in crime rates in the city. An intuitive observation suggests that crime tends to rise during the fall season.

A graph of a graph showing the number of months

Description automatically generated with medium confidence

* We're examining the most unsafe hours within a day.

We first extract the hour from the 'Date' column and create a new column named 'Hour\_Day' to store this information. Then, we calculate the number of crimes per hour and plot the variation of crime rates by hour in Chicago for the year 2022. The plot displays the number of crimes on the y-axis and the hour of the day on the x-axis.

**Result:** Crime rates typically decrease from midnight (00:00 hours) to 05:00 hours.

A graph showing a line of crime rates

Description automatically generated

* ANALYSIS:
* **Classifying Crime Hotspots:**

In this step, we create a new dataset for predictions by extracting specific features such as 'Date', 'Block', 'Location Description', 'Domestic', 'District', 'Month', and 'Primary Type'. Additionally, it derives new columns 'Hour', 'Day', and 'Month\_num' by applying functions to the 'Date' column to extract corresponding hour, day of the week, and month in numerical format, respectively.

* **Using Supervised Machine Learning to Predict Crime Hotspots:**

We group the Data Frame by 'Month\_num', 'Day', 'District', and 'Hour', counting the occurrences of each 'Primary Type' of crime. It then sorts by 'District' in descending order. Required columns are selected, including 'Month\_num', 'Day', 'Hour', 'Primary Type', and 'District'. Average number of crimes per month, per day, per district, per hour is calculated and printed.

Based on the feedback provided during the presentation,

Feature engineering is performed to create a new feature named 'Alarm' based on crime rate, where a binary classification is applied:

1. 0 for low alarm categories (<=7 crimes) and
2. 1 for high alarm categories (>7 crimes).

* RESULTS

Using Supervised Machine Learning Algorithms like Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, and Logistic Regression, we first define independent features (X) and the dependent variable (y) for our predictive model. The dataset is split into training and testing sets (75% for training, 25% for testing).

Based on the feedback during the presentation, we define hyperparameters grids for tuning each classifier to improve overall accuracy and F1-scores. GridSearchCV is used to find the best combination of hyperparameters. This process significantly improved accuracy and F1-scores compared to the initial results before hyperparameter tuning.

After training each classifier on the training data, we make predictions on the test set to calculate accuracy. Confusion matrices are generated to assess performance, and classification reports are printed, containing metrics such as precision, recall, and F1-score for each class.

* **Decision Tree Classifier:**

**A screenshot of a computer

Description automatically generated**

* **Random Forest Classifier:**

A screenshot of a computer

Description automatically generated

* **Gradient Boosting Classifier:**

A screenshot of a computer

Description automatically generated

* **Logistic Regression:**

A screenshot of a computer

Description automatically generated

**INTERPRETATION OF RESULTS:**

|  |  |  |
| --- | --- | --- |
|  | After Hyperparameter tuning | Before Hyperparameter tuning |
| Model | Accuracy | Accuracy |
| Decision Tree | **79.57%** | **73.09%** |
| Random Forest | **81.20%** | **78.08%** |
| Gradient Boosting | **82.24%** | **79.88%** |
| Logistic Regression | **76.45%** | **76.31%** |

* Gradient Boosting Classifier achieved the highest accuracy (82.24%), indicating the best overall performance among all classifiers.
* Logistic Regression exhibited the lowest accuracy (76.45%) among the four classifiers, indicating the weakest performance.
* Gradient Boosting Classifier emerged as the top performer, achieving the highest accuracy and F1-scores for both classes, indicating superior precision and recall compared to other classifiers.
* Random Forest Classifier also performed well, showing high accuracy and F1-scores but slightly lower than Gradient Boosting. Decision Tree Classifier performed moderately, showing decent accuracy and F1-scores but falling behind Random Forest and Gradient Boosting. Logistic Regression exhibited the lowest performance among the classifiers, with lower accuracy and F1-scores, particularly for Class 1.

**CONCLUSION:**

In conclusion, the classification task effectively predicts whether a specific district and hour combination corresponds to a low or high level of criminal activity, aiding in the identification and prioritization of crime hotspots.

Among the models evaluated, the Gradient Boosting Classifier demonstrates superior performance, evident from its higher accuracy and F1-scores for both classes. This model's ability to capture intricate data relationships makes it optimal for identifying crime hotspots and allocating resources efficiently. Therefore, the Gradient Boosting Classifier was chosen as the preferred model for this dataset, surpassing alternatives such as the Decision Tree, Random Forest, and Logistic Regression models.