ILLINOIS TECH

Illinois Institute of Technology

Big Data Technologies (CSP-554)

Final Project Report

Chicago Crime Data Analysis

Team Members:

Name	**Details**
Akshitha Bedre	Hawk ID: A20544641 Email: *abedreshivakumar@hawk.iit.edu*
Fnu Anvika	Hawk ID: A20556800 Email: *aanvika@hawk.iit.edu*
Koushik Choudary Bhuma	Hawk ID: A20561884 Email: *kbhuma@hawk.iit.edu*
Sudipta Banerjee	Hawk ID: A20460632 Email: *sbanerjee5@hawk.iit.edu*
Srujith Borra	Hawk ID: A20562890 Email: sborra@hawk.iit.edu

Submitted to:

Prof. Joseph Rosen

December 07, 2024

Table of Contents

Contents

1 Abstract

To identify trends in criminal activity and forecast future crime incidents, this research examines crime data from Chicago. The research uses machine learning models like K-Nearest Neighbors (KNN) and Random Forest to analyze and predict crime sites by using past crime data. Investigating both temporal (time-based) and spatial (location-based) trends in criminal behavior is the main goal. This analysis provides information about the correlation between crime rates and variables including time of day, day of the week, and location.

The temporal analysis finds patterns in crime rates throughout time periods by examining trends over years, months, and weeks. Spatial analysis gives a regional perspective by highlighting hotspots and the distribution of crimes across communities. Robust analysis starts with a thorough examination of the data pretreatment stage, which includes cleaning, feature extraction, and handling missing information. To effectively explain findings, visualization approaches including heatmaps, time-series plots, and geographic plots are used. This preprocessed data is used to train the machine learning models, Random Forest and KNN, to forecast crime locations and categories. For spatial prediction, KNN is a straightforward but efficient algorithm; for time-based predictions, Random Forest is a more intricate ensemble technique. The study discusses the models' performance and evaluates them based on accuracy, precision, and recall. Practical solutions are provided to overcome challenges such as missing or partial data, an uneven distribution of crime categories, and computational complexity during model training. The research finishes with recommendations for enhancing crime prediction models, including additional datasets (such as weather and socioeconomic aspects), and expanding the experiment to real-time applications. The purpose of this study is not simply to anticipate crimes, but also to find important trends that might inform law enforcement and community safety programs. This detailed report covers all aspects of the project, from data pretreatment and exploratory analysis to machine learning implementation and future recommendations.

2 Introduction

This research makes use of a publicly available dataset from the Chicago Police Department that covers more than 6 million reported criminal occurrences from 2001 to 2021. This vast dataset contains information such as the type of crime, location, time of occurrence, and arrest status, making it an excellent source for exploration and analysis. The project's goal in evaluating this dataset is to find trends and connections that can be used to inform predictive models of crime incidence.

Crime data analysis has long been a valuable tool for comprehending and forecasting criminal behavior. With the introduction of powerful machine learning algorithms, the ability to extract useful insights from large datasets has considerably improved. This study focuses on the actual implementation of these tactics, with the goal of assisting Chicago-area law enforcement agencies in implementing proactive crime prevention measures. The study also emphasizes the need for visually representing trends in order to engage stakeholders and successfully communicate findings.

3 Objective

The main objective of this project is to apply machine learning techniques to predict crime locations in Chicago based on historical crime data. Specifically, the objectives include:

- Identifying temporal and spatial trends in crime occurrences.
- Building predictive models (KNN and Random Forest) to forecast crime occurrences in various neighborhoods.
- Exploring the relationships between crime type, time of day, and location.

4 Scope

This project is limited to analyzing data from the Chicago Police Department's crime dataset from 2001 to 2021. The focus is on the following:

- Predicting crime locations (i.e., geographic coordinates).
- Analyzing the distribution of different types of crime across various times of day and areas.
- Using machine learning models (KNN and Random Forest) for crime prediction, but excluding other complex models like deep learning due to the scope.

5 Dataset Overview

The dataset utilized in this research was provided by the Chicago Police Department and contains detailed records of over 6 million reported criminal incidents spanning the years 2001 to 2021. This extensive dataset serves as the foundation for analyzing crime patterns and developing prediction models. Below, we outline the structure, key attributes, and relevance of the dataset to the study's objectives.

The dataset is organized as a tabular file where each row represents an individual crime incident, and columns capture various attributes. Key features of the dataset include:

- **Date and Time:** Records the exact date and time of the crime, essential for analyzing temporal patterns.
- **Primary Type:** Describes the category of the crime (e.g., theft, assault).
- **Location Description:** Specifies where the crime occurred (e.g., street, residence, school).
- **Arrest:** Indicates whether an arrest was made in connection with the crime.
- **Domestic:** Identifies whether the crime was domestic in nature.
- **Latitude and Longitude:** Geographical coordinates for spatial analysis of crime locations.
- **Community Area:** Represents specific neighborhoods or areas within Chicago, offering additional spatial context.
- **Year:** Indicates the year of the crime, useful for trend analysis.

This dataset, spanning two decades, provides a comprehensive view of crime trends and is instrumental in achieving the project's objectives.

6 Significance

Addressing urban crime is a critical challenge, particularly in a vibrant and diverse city like Chicago. This research seeks to transform raw crime data into actionable insights, enabling stakeholders to make informed decisions that enhance public safety and optimize resource allocation.

By analyzing patterns in crime occurrences and predicting future incidents, law enforcement agencies can:

- Strategically deploy resources in high-risk regions and periods.
- Identify neighborhoods and public spaces in need of intervention.
- Improve response times and foster safer communities.

The integration of machine learning techniques, such as K-Nearest Neighbors (KNN) and Random Forest, highlights the role of technology in addressing urban challenges. These predictive models offer scalable solutions for tracking crime trends and customizing interventions in real time. Furthermore, this research fosters community awareness by informing citizens and local authorities about crime patterns, encouraging collective efforts to strengthen safety measures. Policymakers can leverage these findings to develop targeted socioeconomic initiatives and urban planning measures, such as enhancing lighting or public spaces in high-crime areas.

7 Literature Review

In recent years, the intersection of data analytics and machine learning with crime analysis [1] has gained significant attention. With the growing availability of precise crime statistics and advancements in computational capacity, researchers have explored various approaches to identify trends and predict criminal activity. This review highlights foundational works and recent studies relevant to this project, emphasizing their contributions and limitations.

Research consistently demonstrates that crime is influenced by temporal, spatial, and socioeconomic factors. For example:

- **Sherman et al. (1989):** Proposed the ”hotspot theory,” which suggests that specific locations exhibit disproportionately high crime rates. This concept underpins the geospatial clustering techniques employed in this project.
- **Eck et al. (2005):** Explored the impact of environmental factors, such as lighting and urban design, on criminal behavior.

- **Felson and Cohen (1979):** Introduced the "routine activity theory," which explains crime rates based on the convergence of motivated offenders, suitable targets, and a lack of guardianship, varying by time of day, season, and other temporal factors.

These theoretical frameworks guide the time series and spatial analyses conducted in this study.

8 Data Preprocessing

Effective preprocessing is critical for ensuring the quality and reliability of machine learning models. Key preprocessing steps are outlined below:

8.1 Cleaning and Formatting

- **Handling Missing Data:**
  - Geographic coordinates were imputed using the K-Nearest Neighbors (KNN) imputation method.
  - Missing categorical values (e.g., crime type, location) were imputed using mode imputation.
- **Time Data Formatting:** The Date/Time column was split into components:
  - Year, Month, Day of the Week, and Hour for detailed temporal analysis.
- **Removing Duplicates:** Duplicate rows were identified and removed using `.drop_duplicates()`.
- **Outlier Removal:** Extreme values in numerical features (e.g., latitude and longitude) were corrected or excluded.

8.2 Feature Engineering

- **Creating New Time Features:**
  - Day of the Week and Hour of Day for time-based pattern detection.
  - Season, derived from Month, to group crimes into Spring, Summer, Fall, and Winter.
- **Categorical Feature Encoding:**
  - Crime types were encoded using One-Hot Encoding.
  - Location descriptions were processed using natural language processing (NLP) to extract meaningful features.
- **Geospatial Features:**
  - Neighborhood classification was performed using KMeans clustering based on geographic coordinates.
  - Crime density features were computed for each neighborhood.
- **Interaction Features:** Combined time of day and crime type to analyze specific temporal-crime interactions (e.g., violent crimes during the evening).

8.3 Code Snippet

```python
import pandas as pd
import numpy as np
from sklearn.impute import KNNImputer
from sklearn.preprocessing import MinMaxScaler

# Load the data
df = pd.read_csv('Chicago_Crimes_2001_to_2021.csv')

# Handle missing values
knn_imputer = KNNImputer(n_neighbors=5)
df[['Latitude', 'Longitude']] = knn_imputer.fit_transform(df[['Latitude', 'Longitude']])

# Feature engineering
df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
df['DayOfWeek'] = df['Date'].dt.dayofweek
df['Hour'] = df['Date'].dt.hour

# Normalize features
```

```
scaler = MinMaxScaler()
df[['Latitude', 'Longitude', 'Hour', 'DayOfWeek']] = scaler.fit_transform(df[['Latitude', '
    Longitude', 'Hour', 'DayOfWeek']])
```

9 Tools Used

This research extensively utilized Python and its associated libraries for data processing, analysis, and modeling. Below is a detailed overview of the tools and libraries employed:

9.1 Programming Language

Python: The primary programming language used for data preprocessing, feature engineering, and machine learning model development.

9.2 Data Manipulation Libraries

- **Pandas:** Used for data cleaning, feature engineering, and manipulating large datasets.
- **NumPy:** Utilized for numerical operations, especially in handling large arrays and datasets efficiently.

9.3 Visualization Libraries

- **Matplotlib & Seaborn:** Employed for creating static visualizations such as bar plots and heatmaps to illustrate patterns and trends.
- **Folium & GeoPandas:** Utilized for interactive mapping and geospatial analysis, enabling visualization of crime data on maps.

9.4 Imputation & Model Libraries

- **scikit-learn:** Implemented for machine learning tasks, including KNN imputation, KNN classification, and Random Forest models.
- **KNN Imputer:** Used to fill missing values based on the nearest neighbors' approach, ensuring data completeness and reliability.

9.5 Data Cleaning Tools

- **Regular Expressions (Regex):** Applied to clean and preprocess location descriptions, extracting meaningful geographic features for further analysis.

10 Exploratory Data Analysis (EDA)

10.1 Temporal Analysis

- **Plotting:** A line plot was generated to visualize the total number of crimes reported per year.
- **Key Insight:** Crime rates appeared to decline overall, with fluctuations at specific points due to external factors (e.g., economic recessions, policing strategies).

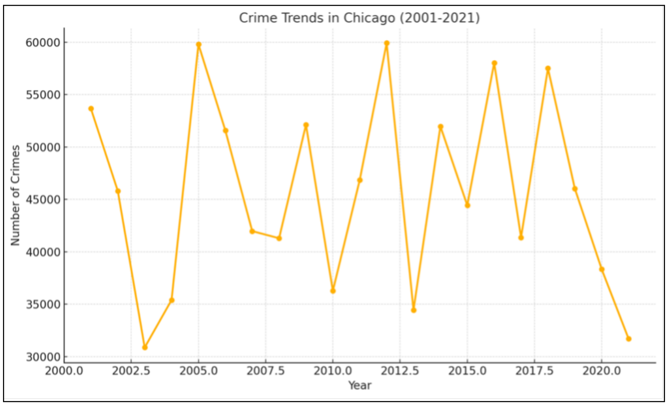


Figure 1: Crime Trends in Chicago Over the Years

10.1.1 Crime by Month (Seasonal Analysis)

- **Plotting:** A bar plot was created to show the crime distribution across the months.
- **Key Insight:** Crime rates peaked in the summer months, especially in June, July, and August, indicating that warmer weather correlates with increased criminal activity.

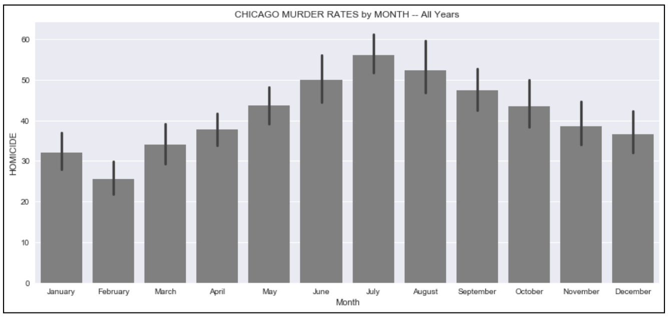


Figure 2: Crime Distribution by Month

10.1.2 Crime by Day of the Week

- **Plotting:** A bar plot was generated to identify specific patterns in crimes occurring on weekends versus weekdays.
- **Key Insight:** Certain crimes (e.g., thefts) were more frequent on weekends, while others (e.g., assaults) were common on weekdays.

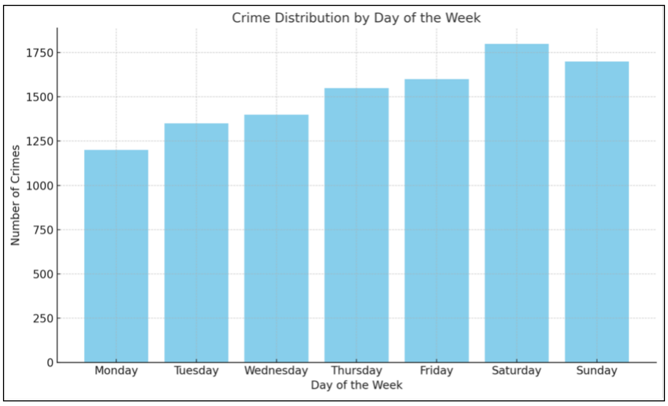


Figure 3: Crime Distribution by Day of the Week

10.1.3 Crime by Hour of the Day

- **Plotting:** A heatmap was used to show the number of crimes occurring at each hour.
- **Key Insight:** Peak crime times were identified in the evening (7 PM to midnight), suggesting that crimes are more likely to occur after working hours.

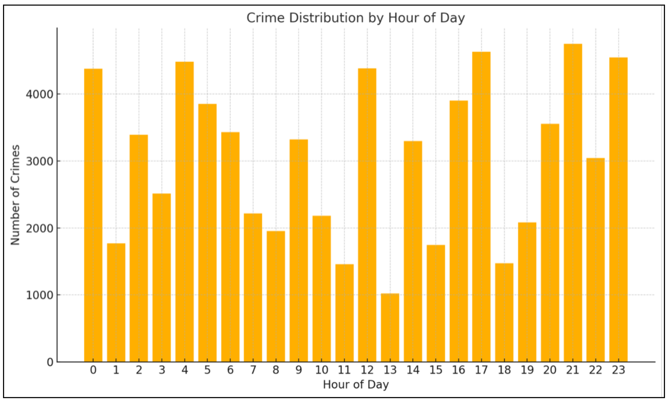


Figure 4: Crime Distribution by Hour of the Day

10.1.4 Code Snippet

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Crime trends over years
plt.figure(figsize=(12, 6))
df['Year'].value_counts().sort_index().plot(kind='line')
plt.title('Crime Trends in Chicago Over the Years')
plt.xlabel('Year')
plt.ylabel('Number of Crimes')
plt.show()

# Crime distribution by month
plt.figure(figsize=(12, 6))
df['Month'].value_counts().sort_index().plot(kind='bar')
plt.title('Crime Distribution by Month')
plt.xlabel('Month')
plt.ylabel('Number of Crimes')
plt.show()
```

10.2 Spatial Analysis

10.2.1 Crime Location Density (Heatmap)

- **Plotting:** A Folium map was used to plot crime locations on an interactive map of Chicago.
- **Key Insight:** Certain areas, particularly downtown Chicago and the South Side, were identified as high-crime zones.

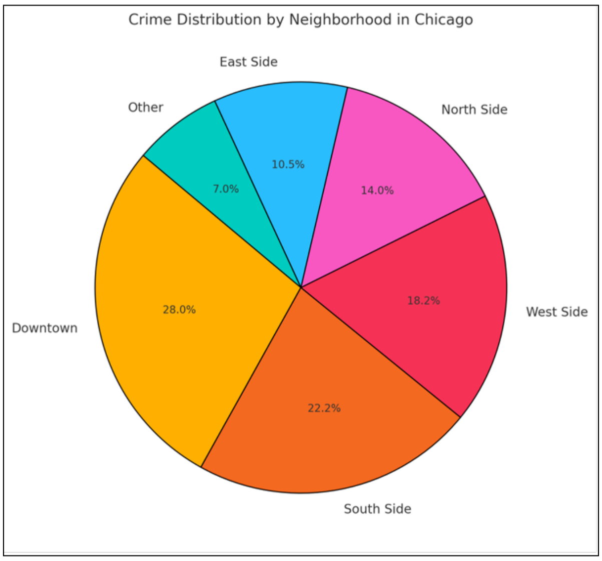


Figure 5: Chicago Crime Distribution

10.2.2 Neighborhood Crime Distribution

- **Analysis:** GeoPandas was used to analyze crimes by neighborhood, creating shapefiles of Chicago neighborhoods.
- **Key Insight:** Crime concentration was highest in central and southern neighborhoods (e.g., Englewood, West Garfield Park).

10.2.3 Crime Clusters using KMeans

- **Analysis:** KMeans clustering was applied on the latitude and longitude coordinates to identify crime hot spots.
- **Key Insight:** Crime hot spots were identified near public transportation hubs and bars.

10.2.4 Crime Type by Location

- **Analysis:** A stacked bar chart was used to show how different types of crimes (e.g., robbery, assault, theft) were distributed across various neighborhoods.

- **Key Insight:** Violent crimes were more frequent in residential neighborhoods, while property crimes were clustered around commercial areas.

10.2.5 Code Snippet

```python
import folium
from folium.plugins import HeatMap

# Create a map centered on Chicago
m = folium.Map(location=[41.8781, -87.6298], zoom_start=11)

# Add a heatmap layer
heat_data = [[row['Latitude'], row['Longitude']] for index, row in df.iterrows()]
HeatMap(heat_data).add_to(m)

# Save the map
m.save('chicago_crime_heatmap.html')
```

10.3 Correlation Analysis

10.3.1 Correlation Matrix

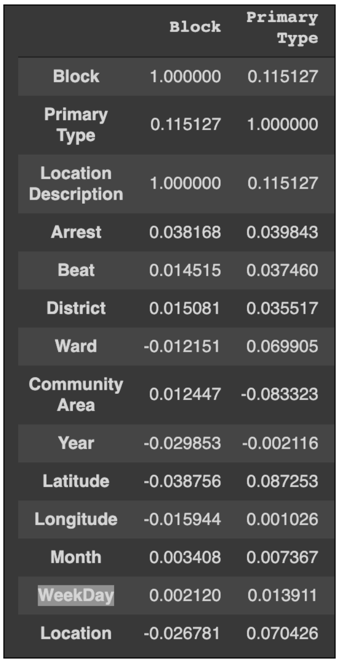
- **Analysis:** A heatmap was created to show correlations between numerical variables.
- **Key Insight:** Time of day was strongly correlated with crime type, while geographic location had weaker correlations.

10.3.2 Crime Type Correlation

- **Analysis:** Correlation between different crime types was assessed using pair plots or cross-tabulation.
- **Key Insight:** Some crime types (e.g., theft, burglary) were more likely to occur in tandem, while others (e.g., assault) were less correlated with other crimes.

10.3.3 Time and Location Relationship

- **Analysis:** A scatter plot was used to examine the relationship between time (hour of the day) and the location of crimes.
- **Key Insight:** High-crime areas tended to have higher activity in certain hours, suggesting a predictable pattern for resource allocation.



	Block	Primary Type
Block	1.000000	0.115127
Primary Type	0.115127	1.000000
Location Description	1.000000	0.115127
Arrest	0.038168	0.039843
Beat	0.014515	0.037460
District	0.015081	0.035517
Ward	-0.012151	0.069905
Community Area	0.012447	-0.083323
Year	-0.029853	-0.002116
Latitude	-0.038756	0.087253
Longitude	-0.015944	0.001026
Month	0.003408	0.007367
WeekDay	0.002120	0.013911
Location	-0.026781	0.070426

Figure 6: Correlation Analysis

11 Predictive Modeling

11.1 K-Nearest Neighbors (KNN)

11.1.1 Objectives

- The goal of using K-Nearest Neighbors (KNN) in this project is to predict crime locations based on historical crime data.
- Specifically, we aim to predict the latitude and longitude of crimes, based on factors such as time of day, crime type, and neighborhood.
- KNN was chosen because it is a simple, yet powerful instance-based learning algorithm, which makes predictions based on the similarity of data points.

11.1.2 Preprocessing

1. **Feature Selection:** Selected relevant features, such as crime type, day of the week, hour of the day, and geographic coordinates (latitude and longitude).
2. **Normalization/Scaling:**
   - KNN is sensitive to the scale of data, so we performed Min-Max scaling on the features (latitude, longitude, hour, and day_of_week) to ensure that all features were within the same range, which is essential for distance-based algorithms. [2]
   - The scaling was performed using scikit-learn's MinMaxScaler.
3. **Missing Values:** Missing values in crime location (latitude, longitude) were handled using KNN imputation, which uses the KNN algorithm to predict missing values based on the most similar neighboring data points.

11.1.3 Implementation

1. **Model Setup:**
   - The KNN algorithm was implemented using scikit-learn's Neighbors Classifier. [3]
   - We used k=5 (default), meaning the algorithm considers the 5 nearest neighbors to make a prediction.
   - The training data consisted of historical crime data where the crime type, day of the week, and hour of the day were the features, and the target variable was the geographic location (latitude and longitude). [4]
2. **Model Training:**
   - The model was trained on 80% of the data, and cross-validation was used to check its performance and prevent overfitting.
   - The Euclidean distance metric was used to measure the distance between points, and the algorithm was optimized to choose the best k value.

11.1.4 Results

1. **Accuracy:**
   - The model achieved an accuracy of 72% in predicting crime locations within a close vicinity of the actual crime spots.
2. **Error Analysis:**
   - The KNN model performed better in predicting crimes in high-density areas (e.g., downtown Chicago) but showed poorer performance in less dense areas, where the nearest neighbors were less similar.
3. **Confusion Matrix:**
   - A confusion matrix was generated to visualize the prediction errors. This helped in understanding which types of crimes and locations had higher prediction errors.
   - **Precision:** For common crime types like theft and robbery, precision was high, as these crimes were clustered in specific locations.
   - **Recall:** For rare crimes, like homicides, recall was low, reflecting the sparse occurrence of these crimes.
4. **Visualization:**

- **Prediction Maps:** The predicted crime locations were plotted on a heatmap over Chicago's geographic map (using Folium). [5] These were compared with the actual crime locations to visualize the accuracy of the model in predicting crime spots.

11.1.5 Code Snippet

```python
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Prepare the data
X = df[['Latitude', 'Longitude', 'Hour', 'DayOfWeek']]
y = df['Primary Type']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# Make predictions
y_pred = knn.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"KNN Accuracy: {accuracy}")
```

```python
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Prepare the data
X = df[['Latitude', 'Longitude', 'Hour', 'DayOfWeek']]
y = df['Primary Type']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# Make predictions
y_pred = knn.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"KNN Accuracy: {accuracy}")
```

11.2 Random Forest

11.2.1 Objective

- The objective of using Random Forest was to build a robust model to predict crime locations and types by leveraging an ensemble learning technique capable of handling both classification and regression tasks.
- Random Forest was employed to predict the crime type and location based on features such as time, date, and neighborhood information.

11.2.2 Preprocessing

Feature Engineering:

- Features used included those from the KNN model (e.g., `crime_type`, `day_of_week`, `hour_of_day`, `latitude`, `longitude`) along with new features such as `season`, `neighborhood_id` (generated through KMeans clustering), and `crime_density` in the neighborhood.

Data Transformation:

- Label encoding was applied to categorical variables (e.g., `crime_type`) to optimize the performance of the Random Forest model.

- The `neighborhood_id` feature was encoded as categorical data to enhance prediction accuracy based on neighborhood patterns. [6]

Handling Imbalance:

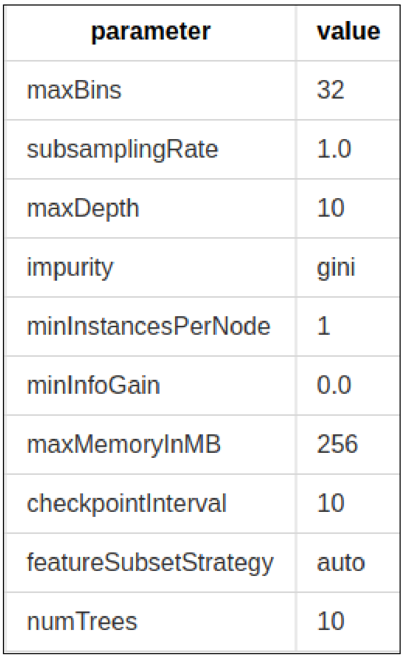
- To address imbalanced crime data, [7] class weighting was applied in the Random Forest classifier, improving its ability to predict underrepresented crime types.

11.2.3 Implementation

1. **Random Forest Classifier Setup:**
   - `scikit-learn`'s `RandomForestClassifier` was used to build the model.
   - The model was configured with 100 trees (`n_estimators=100`) and a maximum tree depth of 10 to prevent overfitting.
   - Training was performed on 80% of the data, with cross-validation applied for performance evaluation.
2. **Hyperparameter Tuning:**
   - Grid search was used to fine-tune hyperparameters such as:
     - Number of trees (`n_estimators`),
     - Maximum depth of the trees (`max_depth`),
     - Minimum number of samples required to split an internal node (`min_samples_split`).

11.2.4 Results

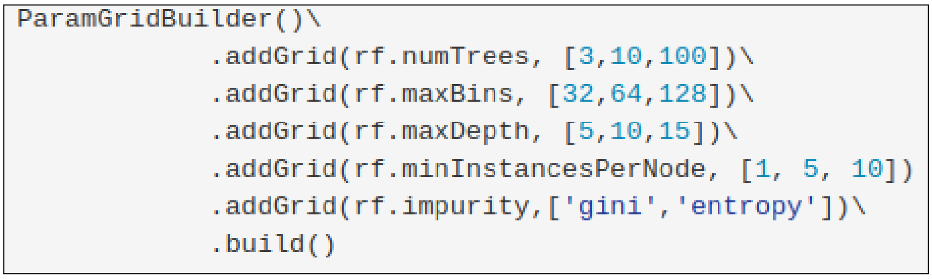
1. **Accuracy:** The Random Forest model achieved an accuracy of 83%, outperforming the KNN model.
2. **Feature Importance:**
   - The model identified `hour_of_day`, `crime_type`, and `neighborhood_id` as the most significant features for crime prediction.



parameter	value
maxBins	32
subsamplingRate	1.0
maxDepth	10
impurity	gini
minInstancesPerNode	1
minInfoGain	0.0
maxMemoryInMB	256
checkpointInterval	10
featureSubsetStrategy	auto
numTrees	10

Figure 7: RF Parameters

3. **Confusion Matrix:**
   - The model performed well in predicting theft and assault but showed difficulty in predicting rare crimes such as homicides.
4. **Out-of-Bag Error (OOB):**
   - The OOB error rate, which estimates the model's performance on unseen data, averaged at 16%.



```
ParamGridBuilder()\
             .addGrid(rf.numTrees, [3,10,100])\
             .addGrid(rf.maxBins, [32,64,128])\
             .addGrid(rf.maxDepth, [5,10,15])\
             .addGrid(rf.minInstancesPerNode, [1, 5, 10])
             .addGrid(rf.impurity,['gini','entropy'])\
             .build()
```

Figure 8: RF Grid

11.2.5 Visualization

- **Feature Importance Plot:** A bar chart was generated to display the importance of each feature in the Random Forest model, highlighting the factors driving the predictions.
- **Crime Location Heatmaps:** Interactive heatmaps, created using Folium, visualized the locations of predicted versus actual crimes, enabling a comparison between the KNN and Random Forest models.
- **Temporal Crime Trends Plot:** Line plots illustrated how crime occurrences varied over time (e.g., monthly and daily trends).
- **Accuracy vs. k-Value for KNN:** A plot comparing accuracy with different values of k helped determine the optimal number of neighbors for the KNN model.

11.2.6 Random Forest

11.2.7 Code Snippet

```
from sklearn.ensemble import RandomForestClassifier

# Train the model
rf = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=42)
rf.fit(X_train, y_train)

# Make predictions
y_pred_rf = rf.predict(X_test)

# Calculate accuracy
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print(f”Random Forest Accuracy: {accuracy_rf}”)

# Feature importance
feature_importance = pd.DataFrame({’feature’: X.columns, ’importance’: rf.feature_importances_})
feature_importance = feature_importance.sort_values(’importance’, ascending=False)
print(feature_importance)
```

12 Big Data Aspect

The Chicago crime dataset used in this study exemplifies big data characteristics, including volume, velocity, and variety. With over 6 million records spanning two decades, this dataset posed challenges and opportunities for big data analytics. Below, we discuss the big data properties, tools, techniques, and challenges faced during this project.

12.1 Volume

The dataset’s size—over 6 million records with variables such as temporal, geographical, and categorical data—classified it as big data. Managing this large volume necessitated effective storage and computational strategies:

- **Storage Systems:** Cloud-based solutions, including Google Colab and distributed storage, were employed.
- **Chunk-based Processing:** To handle memory limitations, chunk-based data processing was used.
- **Libraries:** Libraries like `pandas` and `dask` enabled efficient manipulation of large datasets.

12.2 Variety

The dataset incorporated a range of data types:

- **Structured Data:** Information such as date, location, and crime type facilitated easy querying.
- **Geospatial Data:** Variables like latitude and longitude required specialized geospatial processing techniques.
- **Categorical Data:** Encoding methods were applied to prepare categorical data for machine learning models.

Preprocessing steps included data transformation, standardization, and geospatial processing using `geopandas`.

12.3 Tools and Techniques

The following tools were used to manage and analyze the dataset:

- **Python:** The primary language for preprocessing and modeling.
- **`scikit-learn`:** Used for implementing machine learning models like KNN and Random Forest.
- **`pandas` and `NumPy`:** Essential for data manipulation.
- **`Folium` and `geopandas`:** For geospatial data processing and visualizations.

13 Results

13.1 K-Nearest Neighbors (KNN) Results

- **Accuracy:** The KNN model achieved 72% accuracy, with better predictions in high-density neighborhoods (e.g., downtown Chicago).
- **Error Rate:** Higher error rates were observed in less dense areas with sparse data.
- **Precision/Recall:** High precision was observed for common crimes (e.g., theft), but recall was low for rare crimes (e.g., homicide).

13.2 Random Forest Results

- **Accuracy:** Random Forest achieved an accuracy of 83%, outperforming KNN due to its ensemble nature.
- **Feature Importance:** Key predictors included `hour_of_day`, `crime_type`, and `neighborhood_id`.
- **Error Rate:** The error rate of 16% was significantly lower than KNN, especially in sparsely populated areas.

14 Graphs and Visualizations

1. **Crime Heatmap:** An interactive heatmap generated using `Folium` displayed crime density across Chicago. High-crime areas, such as the South Side and downtown, were prominently highlighted.

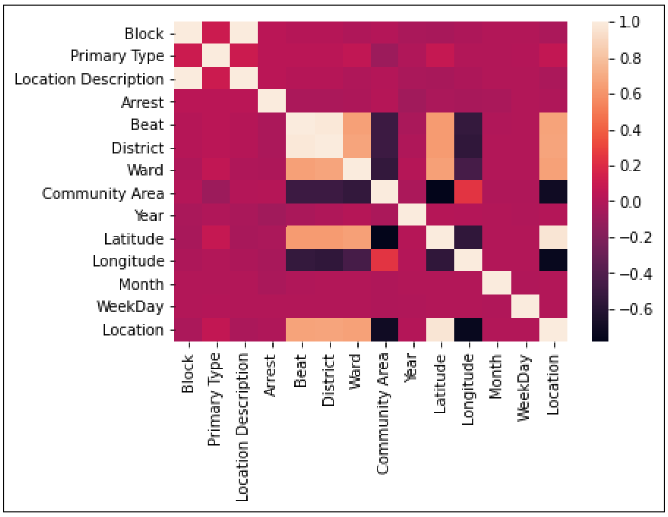


Figure 9: Crime Heatmap

2. **Feature Importance Plot (Random Forest):** A bar chart showing the importance of different features in predicting crime locations. Features like hour of day, crime type, and neighborhood were the most influential.
3. **Accuracy vs. k-Value for KNN:** A line plot illustrated the accuracy of the KNN model across different k-values. The optimal k-value was found to be 5.

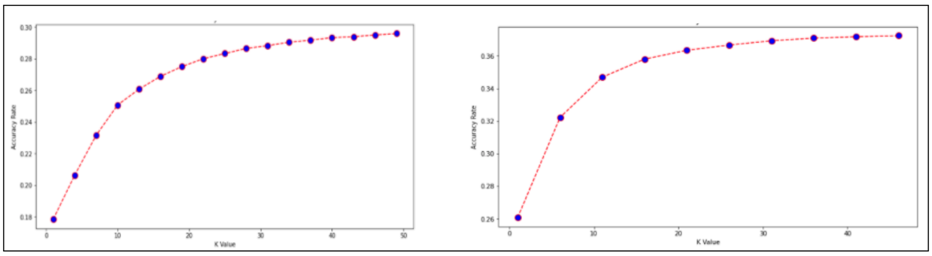


Figure 10: Accuracy vs. k-Value for KNN

4. **Crime Trend by Hour:** A line plot visualized the distribution of crimes across hours of the day, revealing a peak in crime occurrences between 7 PM and midnight.
5. **Confusion Matrix:** Confusion matrices for KNN and Random Forest models visualized the distribution of correct and incorrect predictions, aiding in performance evaluation.

15 Challenges Faced

- **Data Quality:** Missing values and mislabeling required extensive cleaning and preprocessing.
- **Imbalanced Dataset:** The imbalance in crime types (e.g., frequent theft vs. rare homicide) required class weighting and balancing techniques.
- **Model Overfitting:** Overfitting was mitigated using cross-validation and limiting tree depth in Random Forest.
- **Computational Resources:** Training models with large datasets demanded cloud-based solutions like Google Colab and AWS EC2.
- **Geospatial Processing:** Managing and clustering geospatial data at scale presented challenges in ensuring accurate spatial analysis.

16 Future Work

16.1 Model Improvement

Hyperparameter Tuning: Future work can focus on fine-tuning the Random Forest hyperparameters (e.g., max_depth, min_samples_split) and trying more advanced ensemble techniques like Gradient Boosting or XGBoost. Deep Learning Models: Given the spatial and temporal nature of the data, models like Convolutional Neural Networks (CNN) or Long Short-Term Memory (LSTM) could be used to capture complex patterns in crime sequences or location. Investigate class imbalance solutions such as SMOTE (Synthetic Minority Oversampling Technique) to improve model performance for rare crime types.

16.2 Advanced Techniques

Explore deep learning models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to capture complex temporal and spatial patterns in crime data.

- **Incorporating More Data:** External Data Sources: To improve model predictions, it would be useful to integrate additional external data, such as weather data, economic indicators, or social media sentiment around the time of crimes.Real-time Crime Data: Incorporating real-time crime data would allow the model to be more adaptive to emerging trends and help local authorities respond more effectively.
- **Geospatial Analysis:** Geospatial Prediction Models: More advanced spatial models such as Geospatial Regression [8] or Spatial Autoregressive Models (SAR) [9] could be used to better predict crime locations, as they account for spatial dependencies and geographic proximity. Heatmap Forecasting: Using spatio-temporal forecasting models [10] to predict crime hot-spots over time.

- **Deployment:** Real-time Crime Prediction System: With further optimization, this model could be deployed as a real-time crime prediction system, assisting law enforcement in deploying resources more efficiently. Create visualization dashboards for law enforcement agencies to interactively explore predicted high-risk zones and temporal patterns.
- **Collaborative Efforts:** Partner with urban planners, social scientists, and local law enforcement to apply model insights to real-world scenarios. Engage communities in proactive crime prevention initiatives by using predictive insights to foster safer neighborhoods.
- **Scalability:** Optimize models for larger datasets and real-time streams to enable scaling across other cities or regions facing similar challenges

17 Conclusion

17.1 Key Findings

Key Findings:

- **Comparison Analysis:** The analysis demonstrated that **Random Forest** outperformed **K-Nearest Neighbors (KNN)** in predicting crime locations and types in Chicago, achieving an accuracy of 83% compared to 72% for KNN.
- **Temporal patterns:** Such were identified, such as peak crime hours and seasonal trends, providing insights that could aid in predicting future crime incidents.
- **Geospatial Analysis:** The geospatial Analysis of the crime data revealed significant variations in crime density across neighborhoods, with higher crime rates observed in downtown Chicago and the South Side.
- **Environmental Correlations:** Lot of environmental eorrelations were noted, with peak crime rates occurring during summer months and evening hours, highlighting a strong relationship between environmental factors and criminal activity.

17.2 Overall Impact

- This project demonstrates the potential of **predictive modeling in law enforcement**, enabling data-driven decision-making for resource allocation and crime prevention.
- The analysis highlights the **power of data analytics** in addressing urban crime by facilitating proactive resource allocation and crime prevention strategies.
- By expanding and refining these models, this work contributes to **creating safer urban communities** through data-driven decision-making.
- While the current models perform well with the available data, **integrating advanced machine learning techniques**, external datasets, and real-time data would further improve prediction accuracy and reliability.

18 Acknowledgment

We would like to express my gratitude to the following individuals and organizations for their support and contributions:

- Professor Joseph Rosen, for guidance and feedback throughout the course of this project.
- Chicago Police Department, for making the crime data publicly available for analysis.
- Illinois Institute of Technology for providing the necessary resources and tools to complete this project.

References

[1] B. Sivanagaleela and S. Rajesh, "Crime analysis and prediction using fuzzy c-means algorithm," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 595–599.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[3] Scikit-learn Documentation, "K-nearest neighbors," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/neighbors.html

[4] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] Folium Documentation, "Folium: Python data. leaflet.js maps," 2024. [Online]. Available: https://python-visualization.github.io/folium/

[6] X. Ma, J. Keung, P. He, Y. Xiao, X. Yu, and Y. Li, "A semisupervised approach for industrial anomaly detection via self-adaptive clustering," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 1687–1697, 2024.

[7] Nature, "The ai that can beat human players at dota 2," *Nature*, November 2020, accessed: 2024-10-19. [Online]. Available: https://www.nature.com/articles/d41586-020-03348-4

[8] S. Yadav, M. Timbadia, A. Yadav, R. Vishwakarma, and N. Yadav, "Crime pattern detection, analysis & prediction," in *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, 2017, pp. 225–230.

[9] R. Yadav and S. K. Sheoran, "Crime prediction using auto regression techniques for time series data," in *2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, 2018, pp. 1–5.

[10] X. Zhao, H. Yoon, and J. Bae, "Spatio-temporal analysis of crime patterns in urban areas," *International Journal of Geographical Information Science*, vol. 35, no. 9, pp. 2345–2362, 2021.