CrimeWatch : Safety Recommendation

ON

Submitted in partial fulfillment of the requirements of
the degree of

Bachelor of Engineering

(Information Technology)

By

Rakshit Sharma (49)

Veydant Sharma (50)

Avan Shetty (52)

Under the guidance of

Dr. Ravita Mishra

Department of Information Technology

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY,
Chembur, Mumbai 400074

(An Autonomous Institute, Affiliated to University of Mumbai) April 2024

Vivekanand Education Society's
Institute of Technology
(Autonomous Institute Affiliated to University of Mumbai, Approved by AICTE & Recognised by Govt. of Maharashtra)
NAAC accredited with 'A' grade

Certificate

This is to certify that project entitled
"Website Anomaly Detection"
Group Members Names

Mr. Rakshit Sharma (49)
Mr. Veydant Sharma (50)
Mr. Avan Shetty (52)

In fulfillment of degree of BE. (Sem. VI) in Information Technology for Project is approved.

Dr. Ravita Mishra
Project Mentor

External Examiner

Dr.(Mrs.)Shalu Chopra
H.O.D

Dr.(Mrs.) J.M.Nair
Principal

Date: / /2025
Place: VESIT, Chembur

College Seal

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Rakshit Sharma (49)

- - - - - - - - - - - - - - -

Veydant Sharma(50)

- - - - - - - - - - - - - - -

Avan Shetty (52)

- - - - - - - - - - - - - - -

Abstract

As crime patterns become increasingly complex, data-driven approaches are essential for enhancing public safety. This project introduces a machine learning-based system that analyzes historical crime data to predict crime-prone areas and recommend safer districts. By leveraging Random Forest for crime based safe and unsafe categories and XGBoost for safety recommendations, the system offers both accuracy and interpretability. The dataset undergoes preprocessing steps such as cleaning, encoding, and normalization to prepare it for effective model training. The proposed models successfully identify high-risk regions and suggest safer alternatives, enabling more informed decisions by travelers, citizens, and law enforcement agencies. This approach demonstrates the potential of machine learning in promoting smarter and safer communities

Contents

Chapter 1
Introduction

1.1 Introduction

India has experienced a notable evolution in its crime landscape over the past few years. While general crime rates have marginally decreased from 487.8 per 100,000 in 2020 to 442.2 in 2023 the nature of threats has diversified, especially with the rise of cybercrime due to increased digital transactions. This trend underscores a critical need for systems that ensure public awareness and safety, especially for travelers unfamiliar with local crime patterns. Our project, **CrimeWatch** addresses this issue by leveraging machine learning to assess regional safety and provide travel recommendations accordingly.

1.2 Literature Survey

1. Crime Analysis and Prediction using Machine Learning Algorithms (2022)

Authors: P. Kirubanantham, A. Saranya, and A. Prasath G

The study emphasizes the importance of data preprocessing—a critical step that involves cleaning, normalizing, and preparing crime datasets for analysis. By applying these three algorithms to preprocessed data, the authors claim to achieve highly accurate prediction results that outperform existing models. The abstract positions this work within the broader context of data analytics, noting that machine learning techniques like K-Means and Random Forests are widely used, but this research specifically focuses on KNN, Logistic Regression, and SVM for crime analysis

Advantages :

1. It introduces a new approach to crime prediction and classification using a combination of KNN, Logistic Regression, and SVM.

Limitations:

1. While KNN, Logistic Regression, and SVM are effective, the study does not explore more advanced techniques like Random Forests (mentioned in the abstract's context) or deep learning, which might offer better performance for complex crime datasets..

2. Crime Prediction Model using Three Classification Techniques (2024)

Authors: Abdulrahman Alsubayhin, Muhammad Sher Ramzan, Bander Alzahrani

This research compared the performance of three machine learning algorithms—
LightGBM, Random Forest, and Logistic Regression—for predicting crime likelihood. The LightGBM model outperformed the others, offering superior accuracy, followed closely by Random Forest. The study emphasized the potential of ensemble methods for handling large-scale crime datasets and stressed the need for interpretability in real-world applications

Advantages:

1. Successfully demonstrated the power of ensemble methods like LightGBM and Random Forest for crime prediction.
2. Covered a wide range of models, including linear and non-linear approaches.

.**Limitations:**

1. The study didn't explore deep learning models which could capture complex patterns.
2. Feature-level analysis and preprocessing techniques were not discussed in detail.

1.3 Problem Definition

The NCRB provides detailed crime data, but it's often too complex for the general public to understand. This makes it harder for travelers to assess safety, increasing their risk. With rising cyber crimes targeting tourists unfamiliar with local systems, there's a growing need for a platform that simplifies both physical and digital safety information. Addressing this issue requires developing a platform that aggregates, analyzes, and presents crime data in an accessible manner, empowering travelers to make informed and safe choices. The project aims to predict crime-prone areas using crime datasets and classification algorithms, ensuring safer travel recommendations and information about the crime rate
Link for the dataset :
https://www.data.gov.in/resource/district-wise-crime-under-various-sections-indian-penal-code-ipc-crimes-during-2001-2012

1.4 Objectives

- To analyze crime trends and classify regions based on safety levels.
- To develop a predictive ML model for identifying crime-prone areas.
- To implement and compare models including KNN, Decision Trees, Random Forest, and XGBoost techniques for forecasting crime rates.
- To deploy a user-friendly dashboard via ReactJS and Nodejs for interactive and real-time safety analysis with a server for hosting our model

1.5 Proposed Solution

1. Data Collection: Gather crime statistics (e.g., theft, rape, riots) from reliable sources.
2. Preprocessing: Handle missing data, normalize features, and encode categorical variables.
3. Model Training: Train Decision Tree,KNN, Random Forest, and XGBoost models.
4. Evaluation: Compare models using accuracy, precision, and recall.
5. Deployment: Integrate the best-performing model into a user-friendly recommendation system.

1.6 Technology Used

Programming Language: Python

Libraries: Scikit-learn , XGBoost, Pandas, NumPy

Visualization: Matplotlib, Seaborn

Deployment: Flask (for web application)

Chapter 2: Pre-processing

2.1 Dataset Description

Dataset: CrimeWatch: Safety Recommendation Dataset

Data Description: The dataset contains crime statistics across various districts in India, categorized by State/Union Territory (UT), District, and Year. It includes counts of different types of crimes under the Indian Penal Code (IPC), such as Murder, Rape, Theft, and others, along with a total count of IPC crimes. The dataset spans multiple years, with the provided sample covering 2001 to 2012.

Variable	Data Type	Description
State/UT	object	Name of the State or Union Territory
District	object	Name of the District
Year	int64	Year of the crime data
Murder	int64	Number of murder cases
Attempt to Murder	int64	Number of attempted murder cases
Rape	int64	Number of rape cases
Custodial Rape	int64	Number of custodial rape cases
Kidnapping and Abduction	int64	Total kidnapping and abduction cases
Dacoity	int64	Number of dacoity cases
Robbery	int64	Number of robbery cases
Theft	int64	Total theft cases
Riots	int64	Number of riot cases
Importation of the girls from foreign	int64	Number of cases involving importation of girls from foreign countries
IPC crimes	int64	Total number of IPC crimes

Table 2.1 Data Description

Numerical Attributes:
Year, Murder, Attempt to Murder, Rape, Custodial Rape, Kidnapping and Abduction, Dacoity, Robbery, Theft, Riots, CounterFeiting, Importation of the girls from foreign
Cruelty by husband or his relatives, Dowry Deaths, IPC crimes and many such 16 instances

Categorical Attributes:
Nominal Attributes: STATE/UT, DISTRICT

2.2 Handling Missing Data

	0
STATE/UT	0
DISTRICT	0
YEAR	0
MURDER	0
ATTEMPT TO MURDER	0
CULPABLE HOMICIDE NOT AMOUNTING TO MURDER	0
RAPE	0
CUSTODIAL RAPE	0
OTHER RAPE	0
KIDNAPPING & ABDUCTION	0
KIDNAPPING AND ABDUCTION OF WOMEN AND GIRLS	0
KIDNAPPING AND ABDUCTION OF OTHERS	0
DACOITY	0

CRIMINAL BREACH OF TRUST	0
CHEATING	0
COUNTERFIETING	0
ARSON	0
HURT/GREVIOUS HURT	0
DOWRY DEATHS	0
ASSAULT ON WOMEN WITH INTENT TO OUTRAGE HER MODESTY	0
INSULT TO MODESTY OF WOMEN	0
CRUELTY BY HUSBAND OR HIS RELATIVES	0
IMPORTATION OF GIRLS FROM FOREIGN COUNTRIES	0
CAUSING DEATH BY NEGLIGENCE	0
OTHER IPC CRIMES	0
TOTAL IPC CRIMES	0

Based on the provided dataset sample, there are **no missing** values (all columns are populated with valid data). So the dataset was already pure and we needed to just apply the transformations in the further part

2.3 Handling Outliers

```
Q1 = df['TOTAL IPC CRIMES'].quantile(0.25)
Q3 = df['TOTAL IPC CRIMES'].quantile(0.75)
IQR = Q3 - Q1

low_lim = Q1 - 1.5 * IQR
up_lim = Q3 + 1.5 * IQR

outliers = df[(df['TOTAL IPC CRIMES'] > up_lim) | (df['TOTAL IPC CRIMES'] <
low_lim)]['TOTAL IPC CRIMES']
print(f"Number of outliers in TOTAL IPC CRIMES: {len(outliers)}")
```

```
Outliers for THEFT:
Number of outliers: 928
Outlier values:
6       1122
7       2792
18      1296
21      2057
27      1116
        ...
9006    2215
9007    2126
9008    2892
9009    1171
9011    2352
Name: THEFT, Length: 928, dtype: int64
```

```
Outliers for RAPE:
Number of outliers: 568
Outlier values:
28       871
70       817
71        93
115      888
121      144
         ...
8974     174
8981      92
8982    2046
8997     706
9008     116
Name: RAPE, Length: 568, dtype: int64
```

```
Outliers for RIOTS:
Number of outliers: 767
Outlier values:
26       330
28      3001
46       533
49       402
70      2953
        ...
8964     531
8971     397
8975     416
8976     403
8982    6611
Name: RIOTS, Length: 767, dtype: int64
```

```
Outliers for MURDER:
Number of outliers: 585
Outlier values:
1        151
6        182
8        162
12       157
18       214
         ...
8954     224
8973     174
8974     146
8982    2252
8997     521
Name: MURDER, Length: 585, dtype: int64
```

Strategy: Due to the sizable number of outliers, we adopt a hybrid approach

1. **Cap the most extreme** outliers (e.g., top 1% or values beyond Q3 + 3 * IQR) to mitigate their impact on statistical or machine learning models while preserving their presence in the dataset.
2. **Retain moderate outliers** (e.g., within Q3 + 1.5 * IQR to Q3 + 3 * IQR) as they reflect valid high-crime areas essential for analysis.

```
def cap_extreme_outliers(df, column, extreme_threshold='percentile', percentile=99,
iqr_multiplier=3):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    if extreme_threshold == 'percentile':
        upper_bound = df[column].quantile(percentile / 100)
    else:
        upper_bound = Q3 + iqr_multiplier * IQR
    df[column] = df[column].clip(upper=None, lower=upper_bound)
    return df
```

Recalculate the Outliers once again and then

```
print("\nDataset after capping extreme outliers:")
print(df[features].describe())
```

```
Processing THEFT:
Original number of outliers: 928
Number of outliers after capping: 91
Values capped at: 16792.899200000003

Processing RIOTS:
Original number of outliers: 767
Number of outliers after capping: 91
Values capped at: 2759.2064000000014

Processing RAPE:
Original number of outliers: 568
Number of outliers after capping: 91
Values capped at: 1001.8208000000003

Processing MURDER:
Original number of outliers: 585
Number of outliers after capping: 91
Values capped at: 1589.899200000002
```

Values like 2792 for THEFT or 146 for MURDER, which are significant but not extreme, are retained to preserve information about high-crime districts critical for safety recommendations.

This approach reduces the number of outliers (e.g., from 928 to fewer for THEFT) while ensuring the dataset remains representative of real-world crime patterns.

2.4 Feature Scaling

For machine learning models like XGboost and Random Forest we are classifying based on the voting so the numerical attributes may require scaling.

1. YEAR: Already numerical (int64), but we can normalize it to a range [0, 1] for models sensitive to scale.

```
scaler = MinMaxScaler()
data['YEAR'] = scaler.fit_transform(data[['YEAR']])
```

2. Scaling Numerical Attributes : Numerical attributes like MURDER, THEFT, and TOTAL IPC CRIMES have varying ranges. To ensure compatibility with algorithms like K-Means or Logistic Regression, we can standardize them (mean = 0, standard deviation = 1):

We have successfully cleaned the data for our models and this preprocessed dataset is ready for tasks like classifying the districts by crime patterns, predicting total IPC crimes, or analyzing trends over time

Chapter 3
EDA and Visualization

3.1 Measures of Central Tendency

	THEFT	RIOTS	RAPE	MURDER	ARSON	DOWRY DEATHS	IMPORTATION OF GIRLS FROM FOREIGN COUNTRIES
count	9017.000000	9017.000000	9017.000000	9017.000000	9017.000000	9017.000000	9017.000000
mean	16867.880129	2789.013645	1007.426403	1596.389943	25.219918	20.228901	0.197849
std	1239.365409	405.800183	90.632595	173.116417	98.880282	96.970466	2.415039
min	16787.120000	2755.040000	1000.880000	1584.120000	0.000000	0.000000	0.000000
25%	16787.120000	2755.040000	1000.880000	1584.120000	2.000000	1.000000	0.000000
50%	16787.120000	2755.040000	1000.880000	1584.120000	8.000000	5.000000	0.000000
75%	16787.120000	2755.040000	1000.880000	1584.120000	20.000000	16.000000	0.000000
max	53449.000000	11214.000000	3425.000000	7601.000000	2830.000000	2322.000000	83.000000

Table 3.1 Measures of Central Tendency of Dataset

Skewness:
THEFT: 13.578
RIOTS: 13.59
RAPE: 18.146975
MURDER: 20.07
ARSON : 11.324
DOWRY DEATHS: 15.21
TOTAL IPC CRIMES: 7.95

3.2 Visualization based Analysis

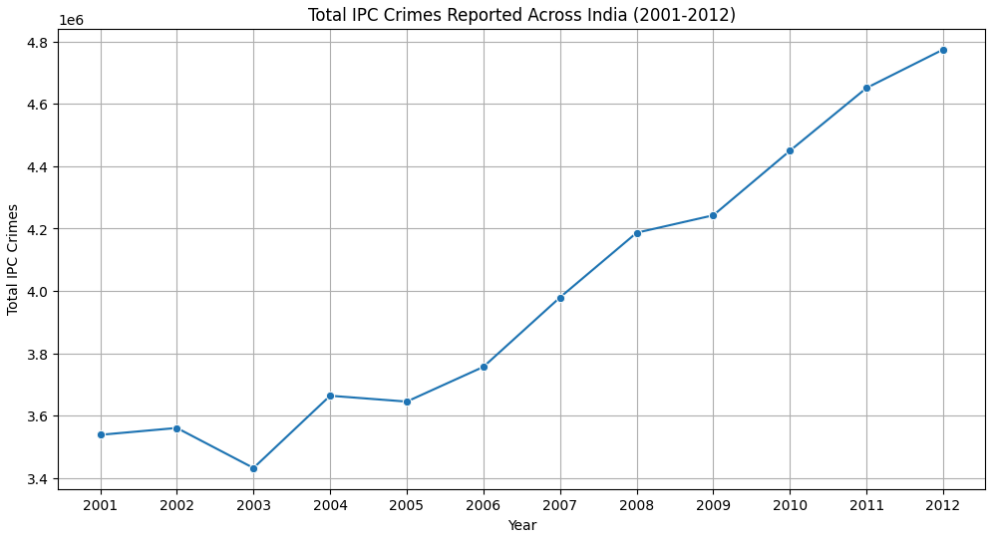


Fig. 3.1 Total IPC Crimes Reported Across India

A dip is observed between 2002 and 2003, followed by a stabilization until 2005, indicating potential fluctuations due to policy changes, socio-economic factors, or data collection

inconsistencies during those years.

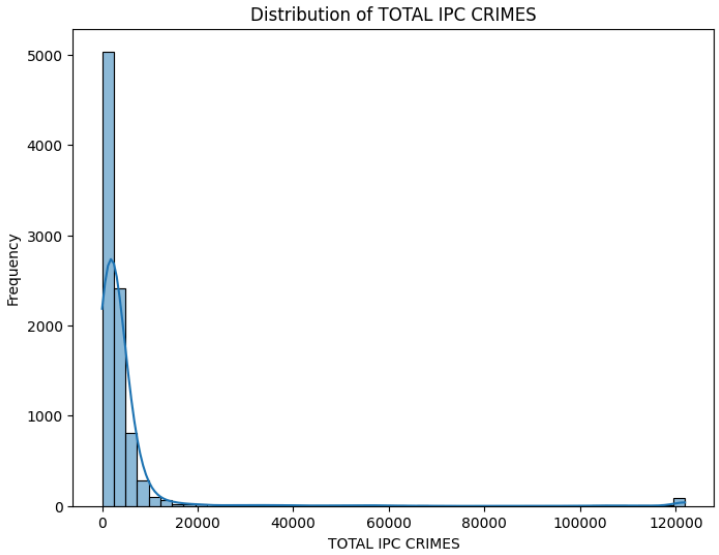


Fig. 3.2 Distribution plot of Total IPC Crimes

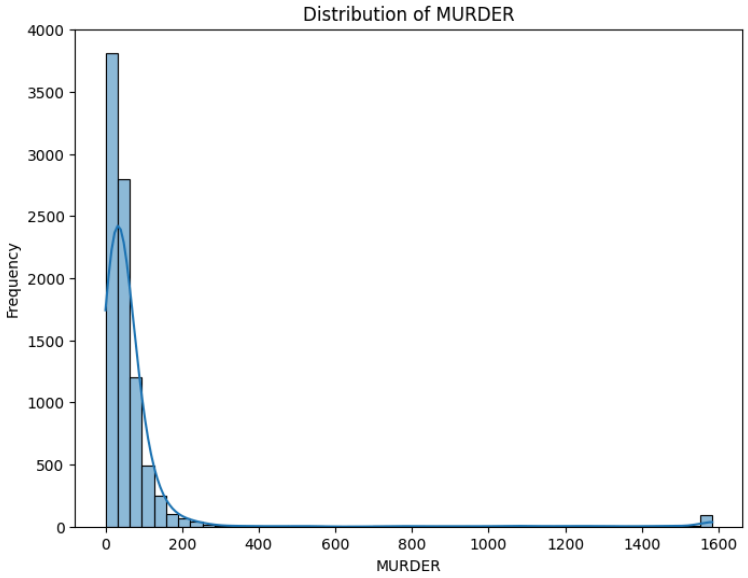


Fig. 3.3 Distribution plot of Murder

This shows that the given data is right skewed for Total IPC Crimes and Murder

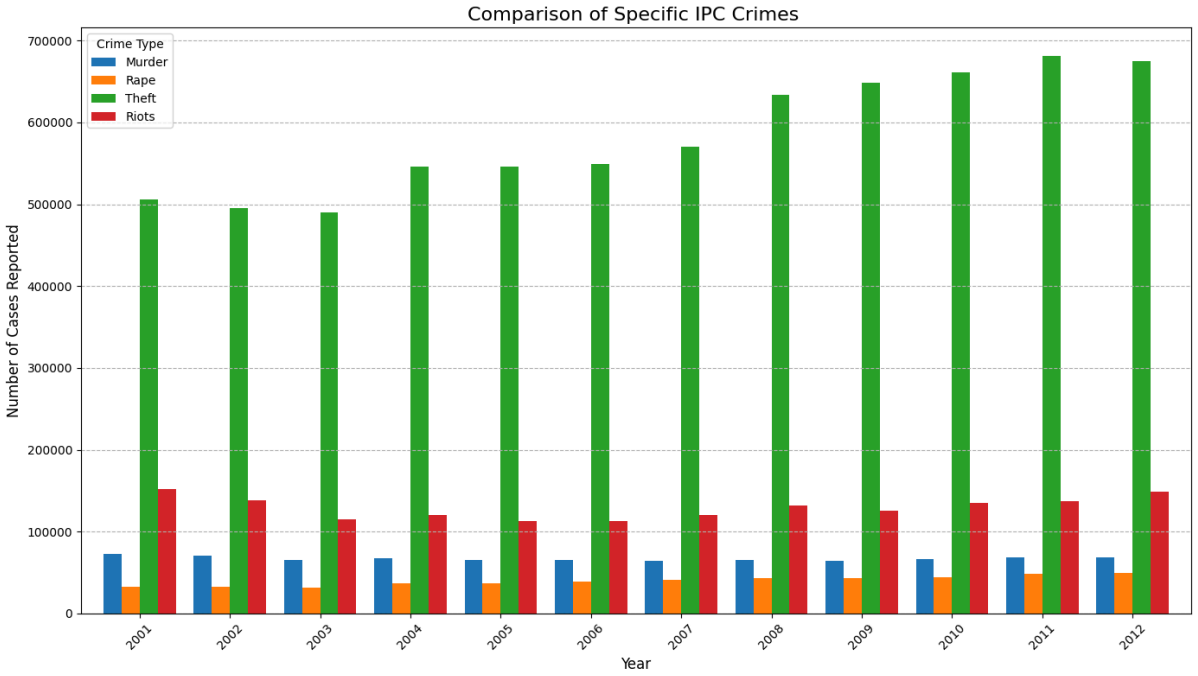


Fig. 3.4 Comparison of Specific IPC Crimes

These graphs show the categorical value distribution of the variables in the dataset.

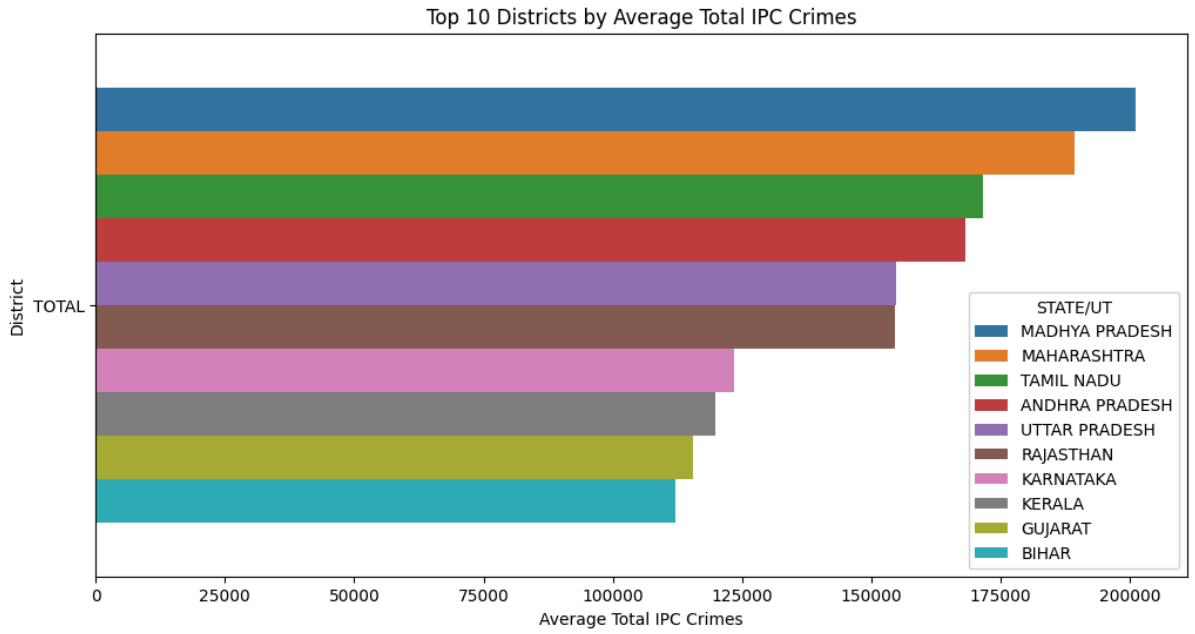


Fig. 3.5 Top districts with most crimes

3.3 Correlation Analysis

1. Relation Between the Theft and the Rape across different years

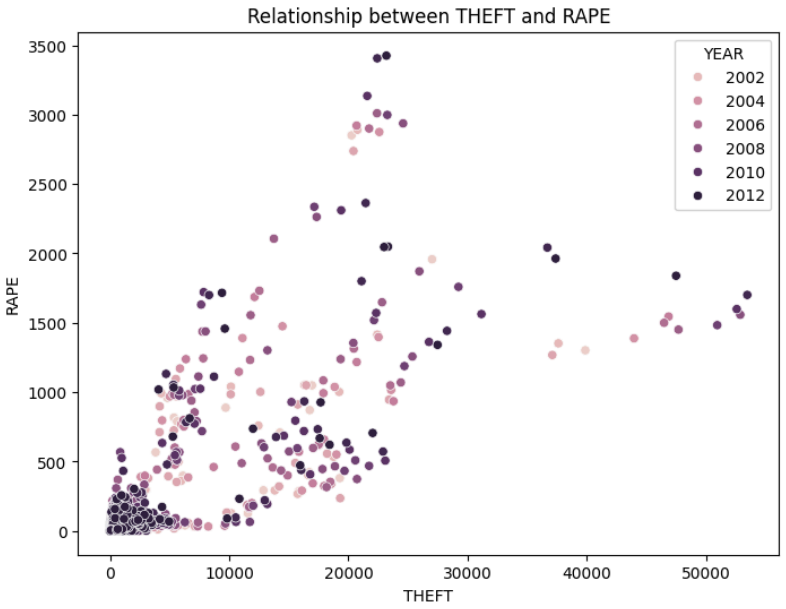


Fig. 3.6 Relation Between the Murder and Theft in different years

Conclusions from Above Graph

1. Higher THEFT counts (up to 50,000) are associated with increased RAPE counts (up to 3500), indicating a positive crime relationship.
2. Points shift toward higher values over time, with 2012 showing more elevated THEFT and RAPE cases than 2002.
3. Some districts exhibit extreme RAPE counts (3000–3500) at moderate THEFT levels, highlighting specific high-crime areas.
4. The density of points increases in later years (e.g., 2010, 2012), suggesting a rise in reported crimes.
5. Early years (2002–2004) show lower and more clustered values, reflecting lower overall crime rates.

2. HeatMaps for relation between all the necessary features of our dataset

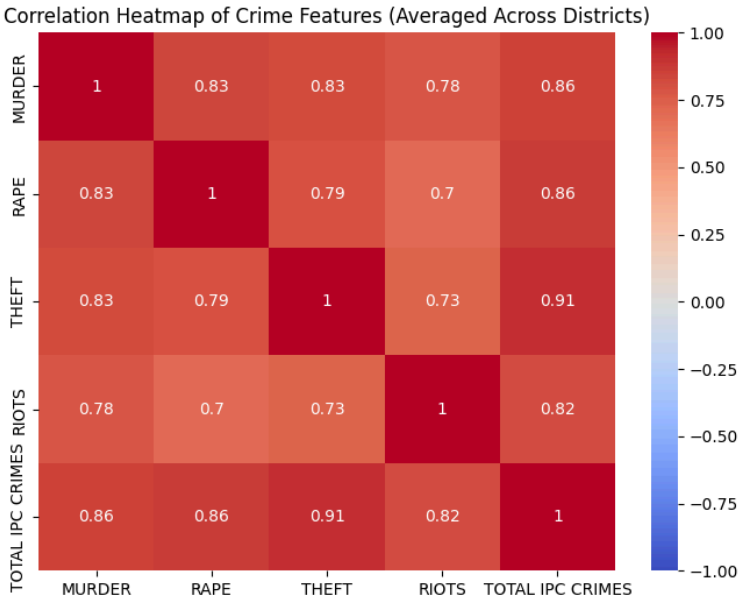


Fig. 3.7 Heatmap of our dataset

Conclusions from the heat map

1. All crime features (MURDER, RAPE, THEFT, RIOTS, TOTAL IPC CRIMES) show strong positive correlations (0.73–1.0), indicating co-occurring crime types.
2. TOTAL IPC CRIMES has the strongest links (0.86–0.91) with individual crimes, acting as a key crime aggregator.
3. Perfect self-correlations (1.0) on the diagonal validate the consistency of averaged district data.
4. THEFT and RAPE correlation (0.79) suggests property and violent crimes often occur together.
5. RIOTS and MURDER correlation (0.78) points to potential socio-economic or conflict-related crime patterns

Chapter 4
Data Modeling

4.1 Flow Of Our System

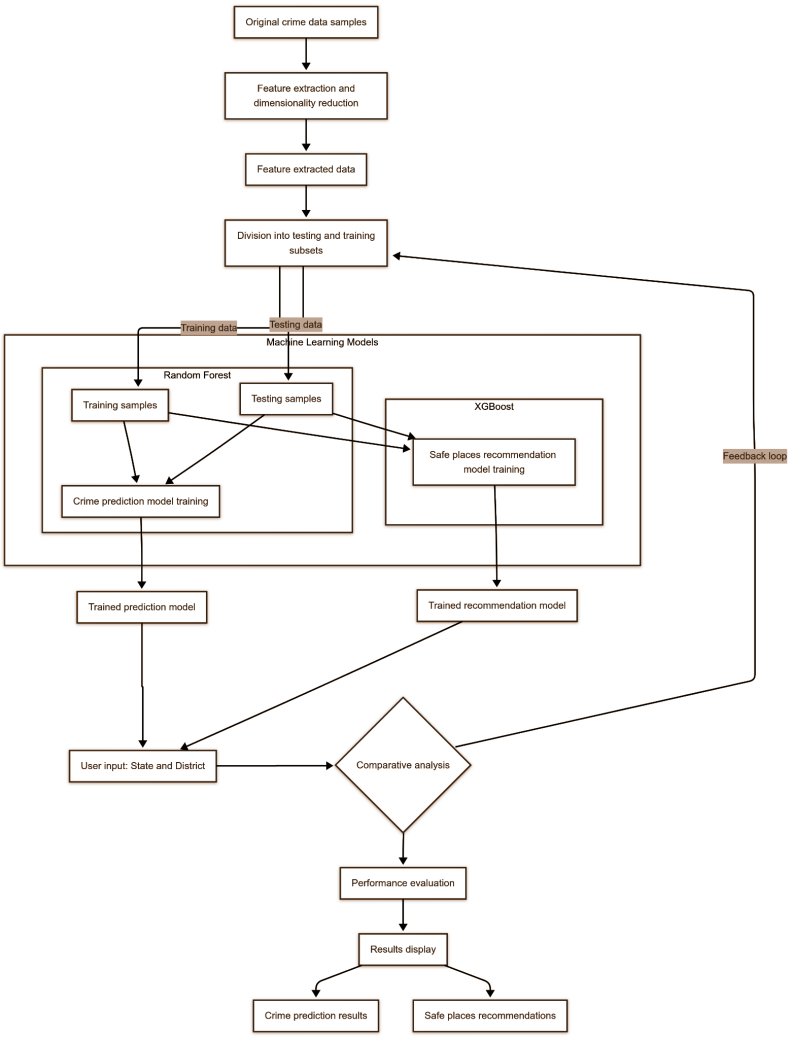


Fig. 4.0 Representation of flowchart

4.2 Model used

Decision Tree

A Decision Tree is a supervised machine learning algorithm used for both classification and regression tasks. It models decisions and their possible consequences in a tree-like structure, where each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents a final predicted class or value.

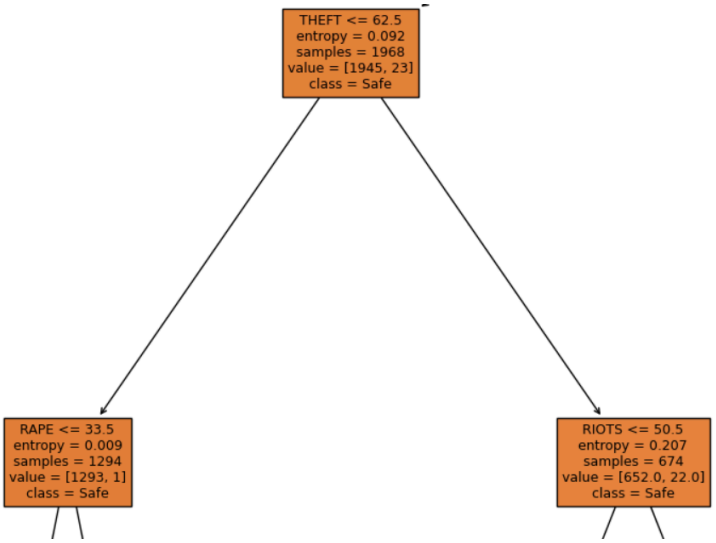


Fig. 4.1 Representation of decision tree

Code:

```
dt_model = DecisionTreeClassifier(
    criterion='entropy',
    max_depth=5,
    min_samples_split=10,
    min_samples_leaf=5,
    random_state=42,
)
dt_model.fit(X_train, y_train)
print("Decision Tree Accuracy:", accuracy_score(y_test, y_pred_dt))
print("\nDecision Tree Classification Report:")
print(classification_report(y_test, y_pred_dt))
```

Output:

```
Decision Tree Accuracy: 0.876940133037694
```

Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

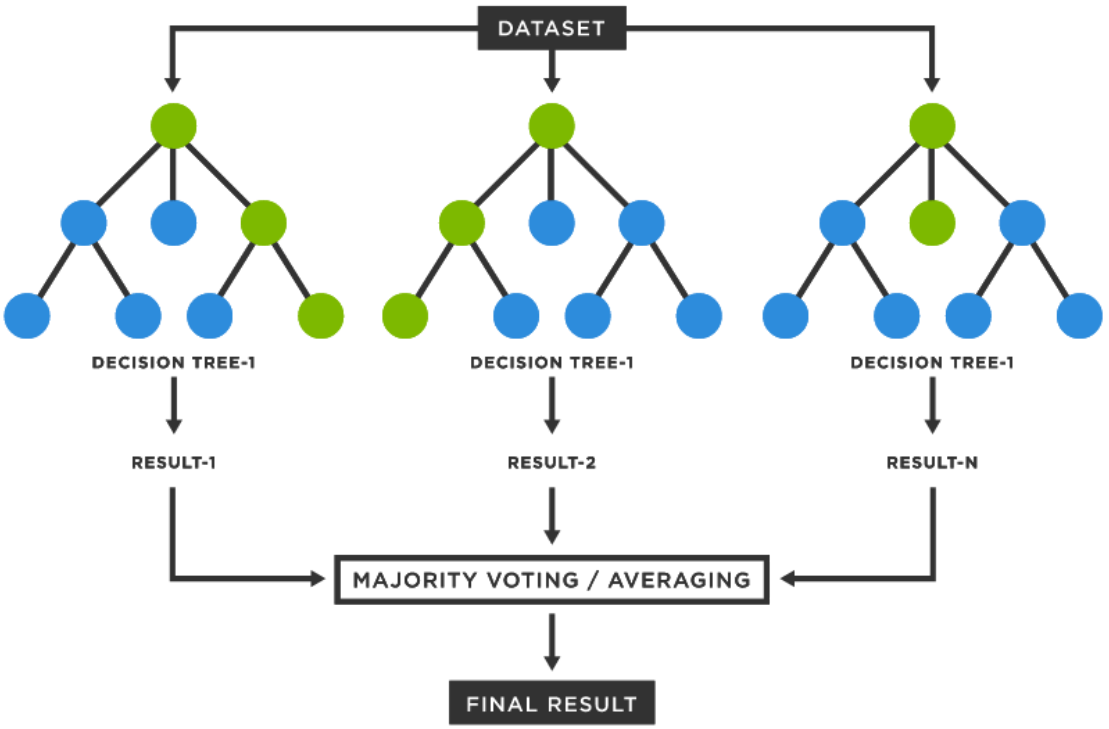


Fig. 4.2 Representation of Random Forest Classification

Code:

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Output:

```
Accuracy: 0.8991130820399114
```

K-Nearest Neighbors

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.

Code:

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
# Train KNN model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

Output:

```
KNN Accuracy: 0.88470066518847
```

XGBoost

XGBoost (Extreme Gradient Boosting) is a powerful, scalable machine learning algorithm based on gradient-boosted decision trees. It optimizes performance by combining sequential tree models, correcting errors from previous trees, and incorporating regularization to prevent overfitting. Known for its speed, efficiency, and high accuracy, XGBoost excels in structured data tasks like classification, regression, and ranking, making it a top choice in competitions and real-world applications.

Code:

```
dtrain = xgb.DMatrix(X_train, label=y_train)
  dtest = xgb.DMatrix(X_test, label=y_test)
  params = {
    'objective': 'binary:logistic',
    'eval_metric': 'auc',
    'max_depth': 5,
    'learning_rate': 0.1,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'seed': 42
  }
  model = xgb.train(params, dtrain, num_boost_round=100,
          early_stopping_rounds=10, evals=[(dtest, 'test')])
```

Output:

```
Accuracy: 0.87
```

Summary of Data modeling

Model	**Score**
Decision Tree	87.69401
Random Forest	89.91130
K- Nearest Neighbor	88.470066
XGboost	87

Table 4.1 Summary of Data modeling

Chapter 5 : Result and Analysis

We evaluated multiple machine learning models to classify districts as "safe" (0) or "unsafe" (1) based on features such as THEFT, RIOTS, RAPE, MURDER as they were the most correlated features among all

a. Random Forest Accuracy: 0.89

Confusion Matrix:

Classification Report:

```
Random Forest Model Evaluation:
Accuracy: 0.8925
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.89      0.89       903
           1       0.89      0.90      0.89       901

    accuracy                           0.89      1804
   macro avg       0.89      0.89      0.89      1804
weighted avg       0.89      0.89      0.89      1804
```

Random Forest was chosen for classifying districts as "safe" or "unsafe" due to its highest accuracy (0.8991) and consistent performance across the classification report, offering balanced precision (0.90) and recall (0.90) for both classes. Its ensemble nature and feature importance insights (e.g., TOTAL IPC CRIMES as a top predictor) make it well-suited for generalizing across the capped crime dataset.

b. XGBoost Accuracy: 0.87

Confusion Matrix:

Classification Report:

```
XGBoost Model Evaluation (on aggregated data):
Accuracy: 0.8735
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.85      0.87        80
           1       0.87      0.90      0.88        86

    accuracy                           0.87       166
   macro avg       0.87      0.87      0.87       166
weighted avg       0.87      0.87      0.87       166
```

XGBoost was utilized for recommending safe places for travelers, capitalizing on its accuracy (0.8735) and ability to model complex relationships in aggregated data. Its strength

in handling non-linear patterns ensures reliable identification of low-crime districts, providing travelers with actionable safety insights.

Features important for the classification:

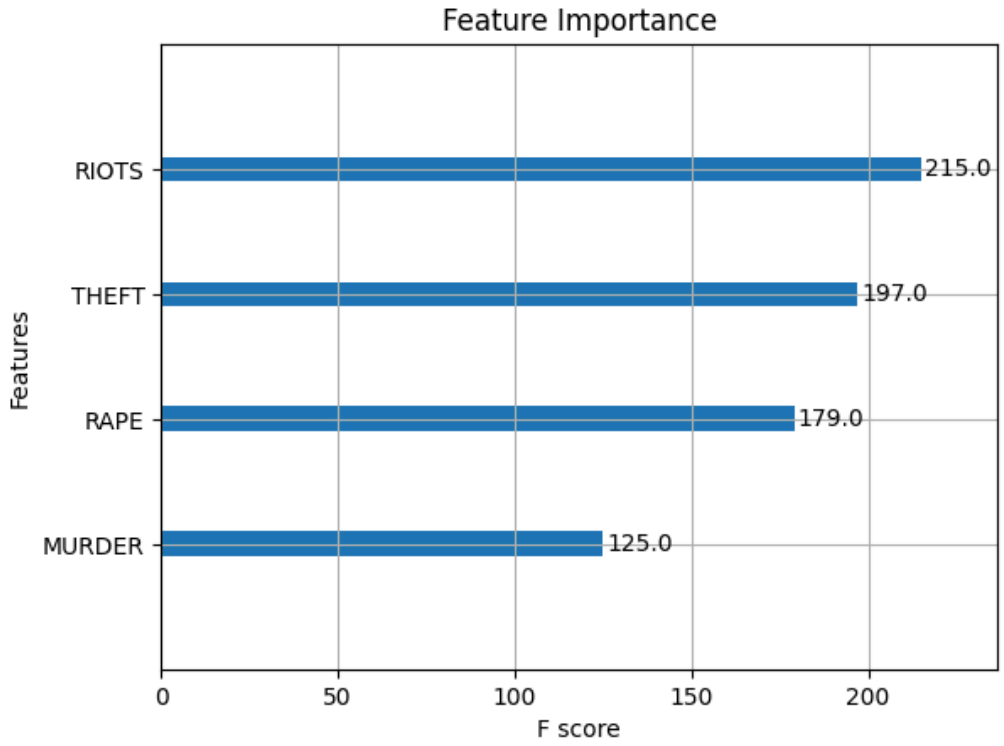
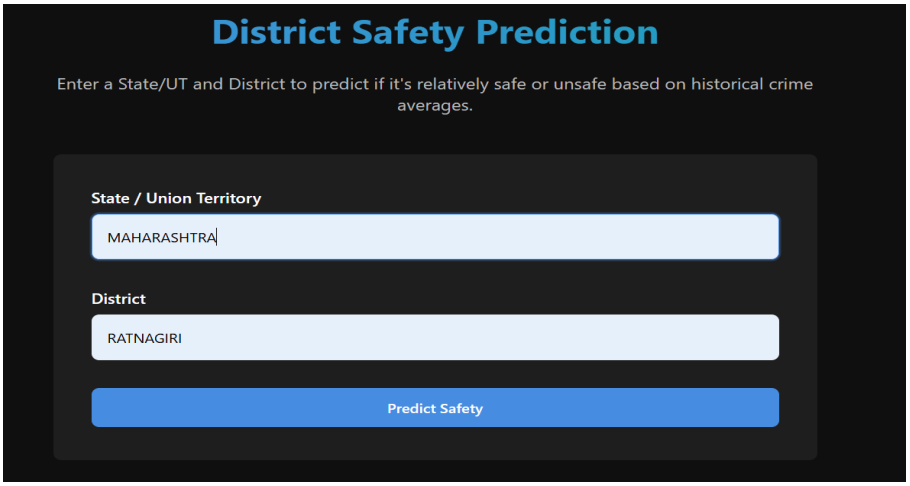


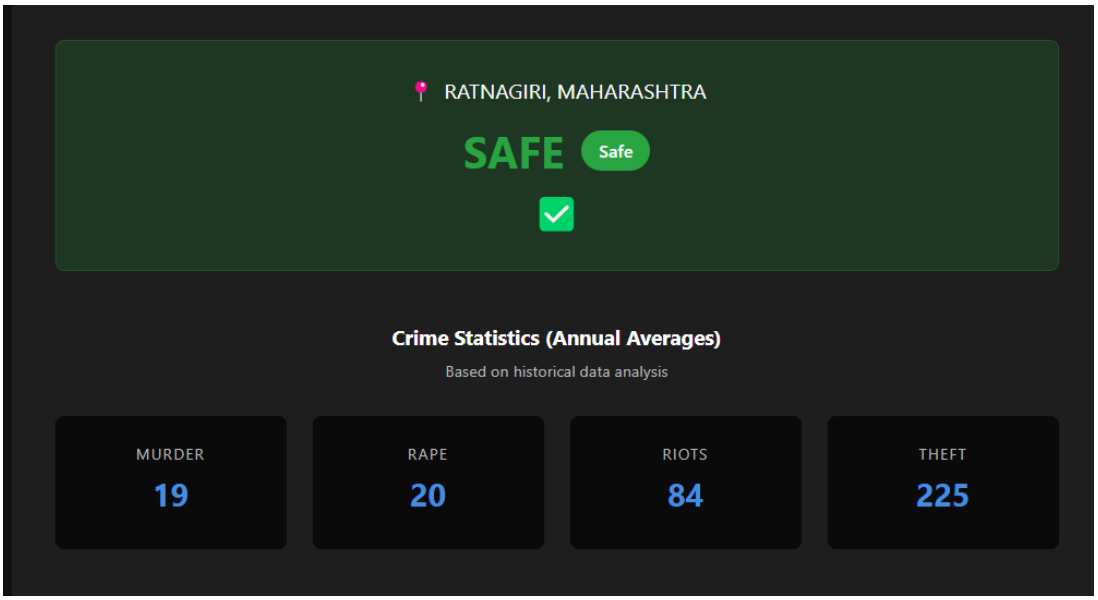
Fig. 6.1 Bar chart showing precedence of importance of features

RIOTS has the highest importance, followed by THEFT and RAPE, while MURDER have the least influence. This suggests that overall crime levels and violent crimes are key indicators for determining district safety.

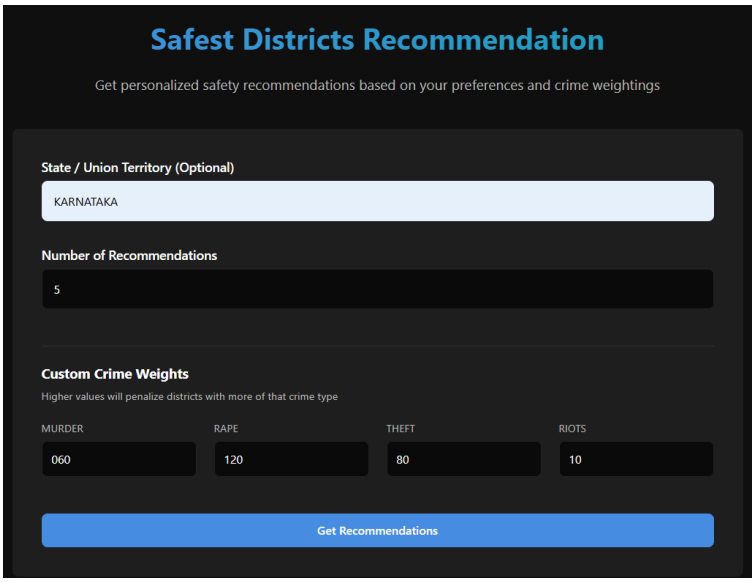
Interactive Dashboard :

1. Classification based in the input of the user the district and state





2. Recommendation of the districts based on the user parameters



Chapter 8 : CONCLUSION

8.1 Conclusion

In this project we demonstrated a practical and data-driven approach to classifying districts as "safe" or "unsafe" and providing safety recommendations for travelers. By leveraging machine learning techniques specifically Random Forest for classification and XGBoost for recommendations—we successfully built a system capable of learning from historical crime patterns to predict district safety and suggest safer travel destinations.The outcome validated the hypothesis that machine learning can play a significant role in enhancing public safety and providing actionable insights for travelers. The models' ability to generalize across districts and their interpretability through feature importance (e.g., TOTAL IPC CRIMES as a key predictor) make them valuable tools for safety planning and policy-making.

8.2 Future Scope

While the current implementation provides a solid foundation for crime data analysis and safety recommendations, several promising directions can be explored to enhance the system's capabilities:

1. **Real-time Crime Monitoring**: Integrating the models into a live system using APIs or data streams could enable real-time safety classification and alerts, allowing for immediate responses to emerging crime trends.
2. **Enhanced Feature Set**: Incorporating additional data sources such as demographic information, economic indicators, or social media sentiment could provide richer context for the models, potentially improving prediction accuracy and recommendation relevance.
3. **Advanced Machine Learning Models**: Exploring more sophisticated architectures like neural networks or ensemble methods could further enhance the models' ability to capture complex crime patterns.
4. **Hybrid Systems**: Combining machine learning with rule-based logic (e.g., crime thresholds set by domain experts) could reduce false positives and adapt to evolving crime patterns while maintaining interpretability.

8.3 Societal Impact

The development and deployment of this crime analysis system have broad and significant implications for society, particularly in the realms of public safety, traveler security, and community trust:

1. **Enhanced Public Safety**: Early identification of high-crime districts can help law enforcement allocate resources more effectively, potentially reducing crime rates and improving community safety.
2. **Traveler Safety**: Providing data-driven safety recommendations for travelers can prevent incidents and promote safer tourism, especially in regions with varying crime rates.
3. **Community Trust**: Transparent crime analysis and safety recommendations can build trust between communities and law enforcement by demonstrating a proactive approach to public safety.
4. **Protection of Vulnerable Populations**: Identifying districts with high rates of specific crimes (e.g., RAPE or DOWRY DEATHS) can help target interventions and support services for vulnerable groups.
5. **Educational Value**: The project serves as a valuable learning tool for students, researchers, and policymakers, promoting awareness and understanding of crime patterns and the application of machine learning in public safety

References

[1] P. Kirubanantham, A. Saranya and A. Prasath G, "Crime Analysis and Prediction using Machine Learning Algorithms," 2022 1st International Conference on Computational Science and Technology (ICCST), CHENNAI, India, 2022, pp. 950-954, doi: 10.1109/ICCST55948.2022.10040319

[2] Alsubayhin, Abdulrahman et al. “Crime Prediction Model using Three Classification Techniques: Random Forest, Logistic Regression, and LightGBM.” International Journal of Advanced Computer Science and Applications (2024): n. pag.

[3] D. Kim et al., "Safe Route Recommendation based on Crime Risk Prediction with Urban and Crime Data," 2023 IEEE Ninth International Conference on Big Data Computing Service and Applications (BigDataService), Athens, Greece, 2023, pp. 111-118, doi: 10.1109/BigDataService58306.2023.00022.