CRIME RATE PREDICTION

A PROJECT REPORT

Submitted by

R.VASHNI BEKASHA

P. SUBHA SRI

M.VEERALAKSHMI

M. SELVA SOBIKA

in partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PET ENGINEERING COLLEGE, VALLIOOR

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2025

BONAFIDE CERTIFICATE

Certified that this project report **"CRIME RATE PREDICTION"** is the bonafide work of **"R.VASHNI BEKASHA (963221104058), P.SUBHA SRI (963221104052), M.VEERALAKSHMI(963221104059), M.SELVA SOBIKA (963221104045)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.BABU RENGARAJAN	**Mrs.J.UMA DEVI**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	**ASSISTANT PROFESSOR**
Department of CSE	**Department of CSE**
PET Engineering College	**PET Engineering College**

Submitted for project **VIVA - VOCE** Examination held at **PET ENGINEERING COLLEGE,** Vallioor on …….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We wish to convey our thanks to our management and the trust members who have provided all the facilities to develop our project successfully.

We are extremely grateful to our beloved and honorable principal **Dr. K. MADHAN KUMAR** for allowing us to have the maximum use of college facilities to do this project.

We are very thankful to Head of the department of Computer Science and Engineering **Dr. S. BABU RENGARAJAN** for supporting us to carry out this project with his moral support and Encouragement.

We extend our sincere thanks to our project coordinator **Mrs. J. ATHISAYA JENSI** who had motivated us and provided guidance throughout the course of our project.

We are indebted to our project supervisor **Mrs. J. UMA DEVI** for her guidance and helpful comments during the course of our project.

We are very thankful to all the teaching and non-teaching staff members of Computer Science and Engineering Department for their timely help.

Finally, we are thankful to our beloved parents and friends for their moral support

ABSTRACT

Crime is a major problem in today's world, and it is a threat to global security. The population of cities is constantly increasing, resulting in an increase in crime rates. Officials are tasked with the monumental challenge of accurately predicting future crime rates and attempting to reduce them. To help in this regard, various large datasets have been reviewed, extracting information such as location and crime type. Crime prediction utilizes various methods to identify areas that are likely to experience higher levels of crime. Given a set of historical crime data, develop a predictive model to identify areas of high risk for future criminal activity, to improve the accuracy of crime prevention and policing efforts. These methods include analyzing past crime data, identifying crime hotspots, and utilizing predictive analytics. The main aim of this project is to develop a system that can accurately predict crime rates and identify potential future crime trends. This information can then be used by officials to devise strategies to reduce crime rates and create a safer environment. To predict the crime rate (dependent variable) based on the year, location, and type of crime (independent variables), various types of machine learning algorithms will be applied. The system will examine how to convert the crime information into a regression problem, thus helping the officials to solve crimes faster. Crime analysis using available information to extract patterns of crime. Based on the territorial distribution of existing data and the recognition of crimes, various multi-linear regression techniques can be used to predict the frequency of crimes.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	iv
1	**INTRODUCTION**	**7**
2	**LITERATURE REVIEW**	**8**
3	**SYSTEM ANALYSIS**	**11**
	3.1 EXISTING SYSTEM	11
	3.2 PROPOSED SYSTEM	11
4	**SOFTWARE REQUIREMENTS**	**12**
	4.1 PYTHON 3.x	12
	4.2 JUPYTER NOTEBOOK	12
	4.3 PANDAS	13
	4.4 NUMPY	13
	4.5 MATPLOTLIB	13
	4.6 SEABORN	14
	4.7 SCIKIT-LEARN	14
	4.8 JOBLIB	15
	4.9 WEB BROWSER	15
5	**SYSTEM REQUIREMENTS**	**16**
	5.1 HARDWARE REQUIREMENTS	16
	5.2 SOFTWARE REQUIREMENTS	16
6	**SYSTEM ARCHITECTURE**	**17**

7	**SYSTEM STUDY**	**21**
	7.1 FEASIBILITY STUDY	21
	7.2 TECHNICAL FEASIBILITY	21
	7.3 ECONOMIC FEASIBILITY	21
	7.4 OPERATIONAL FEASIBILITY	21
8	**SYSTEM TESTING**	22
	8.1 TYPES OF TESTS	22
	8.1.1 UNIT TESTING	22
	8.1.2 INTEGRATION TESTING	22
	8.1.3 PERFORMANCE TESTING	23
	8.1.4 MODEL TESTING	23
9	**IMPLEMENTATION**	24
10	**COMPARISON CHART**	26
11	**MODULE DESCRIPTION**	28
12	**OUTPUT SCREENSHOT**	**29**
13	**CONCLUSION & FUTURE ENHANCEMENT**	**33**
	12.1 CONCLUSION	33
	12.2 FUTURE ENHANCEMENT	33
14	**APPENDIX**	**34**
15	**REFERENCES**	**41**

1.INTRODUCTION

Crime is an act that is prohibited by law and is punishable by a fine, imprisonment, or other legal action. Every day, reports of criminal activity fill our news outlets and social media platforms, painting a picture of a world in which crime is an ever-present concern. From robberies and violent assaults to cybercrimes and white-collar fraud, there is seemingly no end to the number of ways in which criminals can cause harm. Crime has been a part of human civilization since time immemorial. It has become increasingly prominent in today's world. The rise of technology has created a variety of new crimes, while the emergence of globalization has made the world a smaller place, allowing criminals to move and operate in different countries. Crime is uncertain and cannot be predicted. Crime prediction is significant to determine increase or decrease in crime rate from preceding years. A huge number of crimes happen every second in different places, in different patterns and in different times and the number is increasing each growing day. A good prediction technique provides a more rapid evolution of criminal data sets. It helps in predicting the correct place of crime and criminal activity, as well as aids in keeping track of resources pertaining to the analysis of crime. Crime prediction using machine learning is an emerging field of study that uses sophisticated algorithms and data-driven methods to detect and predict criminal activities. Machine learning algorithms can be used to identify patterns in data that may indicate a future crime, such as past criminal activities, demographic information, and environmental factors. By leveraging such data, machine learning can be used to create predictive models that identify the likelihood of a certain crime occurring in a particular area or time frame. Additionally, machine learning can be used to develop insights into the behavior of criminals, helping law enforcement professionals better understand and address criminal activity.

2. LITERATURE REVIEW

1. CRIME RATE PREDICTION USING KNN

Authors : Ms. Vrushali Pednekar, Ms. Trupti Mahale, Ms. Pratiksha Gadhav

SUMMARY:

This system looks at how to convert crime information into a data-mining problem to help detectives solve crimes faster. It focused on crime analysis, extracting target datasets, data pre-processing, data mining, and interpretation and using discovered knowledge. The proposed model of crime analysis and prediction uses a general algorithm which takes raw data of crime from a government repository as input and produces a correlated dimensions model for crime analysis and prediction as output. It also uses various data mining techniques to predict the frequency of occurring crime based on territorial distribution of existing data. It also involves data cleaning and treating missing values to improve the quality of data for mining. Finally, it provides SQL or reports to interpret the discovered patterns and take actions based on the knowledge.

2. USING FACEBOOK INTERESTS TO IMPROVE PREDICTIONS OF CRIME RATES IN URBAN AREAS

Authors : Masoomali Fatehkia, Dan O'Brien, Ingmar Weber

SUMMARY:

This article focused on the potential for using data from the Facebook Advertising API to gain insight into the distribution of individual-level processes in relation to crime rates across different neighborhoods. It begins by describing existing theories of criminogenesis related to factors such as poverty, social disorganization, income inequality, and impulsivity. It then outlines how the API could be used to measure the

prevalence of interests among a ZIP code's Facebook population, which can be used to reflect behavioral and attitudinal features of a population. The article concludes by noting that this is an exploratory study, and that the results should be interpreted as predictions of reported crime rates, not necessarily crime itself.

3. A CLUSTERING BASED HOTSPOT IDENTIFICATION APPROACH FOR CRIME PREDICTION AREAS

Authors : Gaurav Hajela, Dr. Meenu Chawla, Dr. Akhtar Rasool

SUMMARY:

The research on crime revealed that it can be represented with a spatio-temporal pattern across geographical space. There are many indicators of crime such as urban or census-based indicators, street light and daylight, social media-based indicators, population flow indicators, and climate-based indicators. A crime hotspot is an area with a higher concentration of crime than the rest of the area. This paper proposes a crime prediction model for the dataset of San Francisco, which includes crime hotspot identification, dataset preparation, and crime prediction approach. The performance parameters used to evaluate the model are discussed and the results are presented.

4. CRIME RATE PREDICTION USING MACHINE LEARNING AND DATA MINING

Authors : Sakib Mahmud, Musfika Nuha, Abdus Sattar

SUMMARY:

This relationship examined in this paper was between crime and different features in the criminology literature. To reduce crime and detect criminal activity, the author used Z-Crime Tools and Advanced ID3 algorithms with data mining technology, K-

Means Clustering and deep learning algorithms, random forest and naïve Bayes algorithms, and multi-linear regression. The author also used Apriori algorithm and Naive Bayes algorithm for identification of criminal trends and patterns.

5. SURVEY ON CRIME ANALYSIS AND PREDICTION USING DATA MINING TECHNIQUES

Authors : Ms. Vrushali Pednekar, Ms. Trupti Mahale, Ms. Pratiksha Gadhave, Prof. Arti Gore

SUMMARY:

This system looks at how to convert crime information into a data-mining problem to help detectives solve crimes faster. It focuses on crime analysis, extracting target datasets, data pre-processing, data mining, and interpretation and using discovered knowledge. The proposed model of crime analysis and prediction uses a general algorithm which takes raw data of crime from a government repository as input and produces a correlated dimensions model for crime analysis and prediction as output. It also uses various data mining techniques to predict the frequency of occurring crime based on territorial distribution of existing data. It also involves data cleaning and treating missing values to improve the quality of data for mining. Finally, it provides SQL or reports to interpret the discovered patterns and take actions based on the knowledge. This proposed system presented a new framework for clustering and predicting crimes based on real data. Considering the methods proposed for crime prediction shows that the parameters such as the effect of outliers in the data mining preprocessing, quality of the training and testing data, and the value of features have not been addressed before. This proposed system predicts crime prone regions in India on a particular day.

3.SYSTEM ANALYSIS

3.1 Existing System:

Crime rate prediction is based on early implementations of crime mapping and analysis systems, such as basic crime data dashboards used by police departments. These systems typically rely on historical crime records and statistical summaries to track crime rates, identify hotspots, and allocate resources accordingly. However, these older systems are limited in their predictive power. They use basic statistical tools and do not adapt dynamically to changing crime patterns. Some systems like Predictive Policing Tools (e.g., PredPol) have attempted to apply data science and machine learning to forecast crime locations based on past trends, but they are often expensive, closed-source, and not widely accessible.

3.2 Proposed System:

The Crime Rate Prediction model that leverages machine learning techniques to analyze historical crime data and predict future crime trends. The main objective of this system is to assist law enforcement agencies and urban planners in understanding crime patterns and enabling proactive decision-making. The system uses a dataset containing information about various types of crimes and their associated features such as location, time, and crime category. Using data preprocessing techniques, the system cleans and transforms raw data into a suitable format for machine learning models. It employs algorithms like Linear Regression and Random Forest Regressor to learn from historical trends and predict the crime rate in different areas.

4. SOFTWARE REQUIREMENTS

4.1 Python 3.x

Definition:

Python is a powerful, high-level programming language with simple syntax, making it ideal for data science, machine learning, and web development. It supports multiple programming paradigms and has a vast collection of libraries and frameworks.

How it's applied in the project:

Python is the foundation of the entire Crime Rate Prediction project. All the code for reading the dataset (National Crime Rate Bureau), cleaning and preprocessing data, training the machine learning models (like Linear Regression and Random Forest), visualizing data, and evaluating performance is written in Python. The simplicity of Python helps developers quickly write and test data-driven models.

4.2 Google colab/Jupyter notebook

Definition:

These are interactive environments that let developers write and execute Python code in a notebook format. Google Colab runs on the cloud and provides free access to GPUs, while Jupyter Notebooks run locally.

How it's applied in the project:

The project code is developed in Google Colab, allowing users to import datasets, run code, and view outputs in real time. Colab also supports sharing, which is helpful for collaboration. Since it runs on Google's servers, there's no need to install Python or libraries locally.

4.3 Pandas

Definition:

Pandas is a Python library used for data manipulation and analysis. It provides two main data structures—Series and Data Frame—that simplify operations like filtering, grouping, and merging datasets.

How it's applied in the project:

Pandas is used to load the crime dataset from a CSV file, check for missing values, clean the data, and prepare it for analysis. Functions like read_csv(), dropna(), and groupby() are used to clean and structure the data before it's fed into machine learning models.

4.4 NumPy

Definition:

NumPy stands for Numerical Python. It is a library for handling large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on them.

How it's applied in the project:

NumPy supports efficient numeric computation in the background. It is used when performing mathematical operations on datasets, especially when converting data into arrays or matrices required by machine learning models.

4.5 Matplotlib

Definition:

Matplotlib is a plotting library in Python for creating static, animated, and interactive visualizations. It’s used to create basic plots such as bar charts, line graphs, and scatter plots.

How it's applied in the project:

It is used to visualize trends in the crime dataset—such as crime rates over time or across regions. Visual representation helps developers and users understand patterns and validate prediction results.

4.6 Seaborn

Definition:

Seaborn is a data visualization library built on top of Matplotlib. It provides a high-level interface for drawing attractive statistical graphics such as heatmaps and boxplots.

How it's applied in the project:

Seaborn is used to create correlation plots and heatmaps to explore the relationships between different features in the dataset. For example, it can help determine how strongly certain features (like area or type of crime) affect the crime rate.

4.7 Scikit-learn

Definition:

Scikit-learn is a widely used Python library for machine learning. It provides easy-to-use tools for data preprocessing, model selection, training, testing, and evaluation.

How it's applied in the project:

This is the main machine learning engine of the project. It is used to:

- Encode categorical features (e.g., using LabelEncoder)
- Split the data into training and testing sets (train_test_split)
- Train models like LinearRegression() and RandomForestRegressor()
- Evaluate model performance using metrics like Mean Squared Error and R^2 Score

4.8 Joblib

Definition:

Joblib is a Python library used to save and load models or large data efficiently. It is especially useful when dealing with machine learning models that are time-consuming to train.

How it's applied in the project:

After the machine learning model is trained, it is saved using joblib.dump() so that it doesn’t need to be retrained every time. The saved model can later be loaded using joblib.load() for making predictions on new data.

4.9 Web Browser

Definition:

A web browser is a software application used to access and interact with content on the internet. Common browsers include Google Chrome, Firefox, and Microsoft Edge.

How it's applied in the project:

A web browser is used to open and run the project on Google Colab and visualize the project outputs .It acts as a user interface for development and testing.

5. SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS:

- Processor: Intel Core i3 (or equivalent)
- RAM: 4 GB
- Storage: 2 GB of free disk space
- Display: 1024×768 resolution
- Input Devices: Keyboard and mouse

5.2 SOFTWARE REQUIREMENTS:

- Operating System: Windows 10
- Programming Language: Python 3.x
- Development Environment: Jupyter Notebook
- Libraries and Packages

  NumPy – for numerical operations

  Pandas – for data manipulation and analysis

  Matplotlib – for data visualization

  Seaborn – for statistical data visualization

  Scikit-learn – for machine learning algorithms

  Joblib – for saving and loading machine learning model

6. SYSTEM ARCHITECTURE

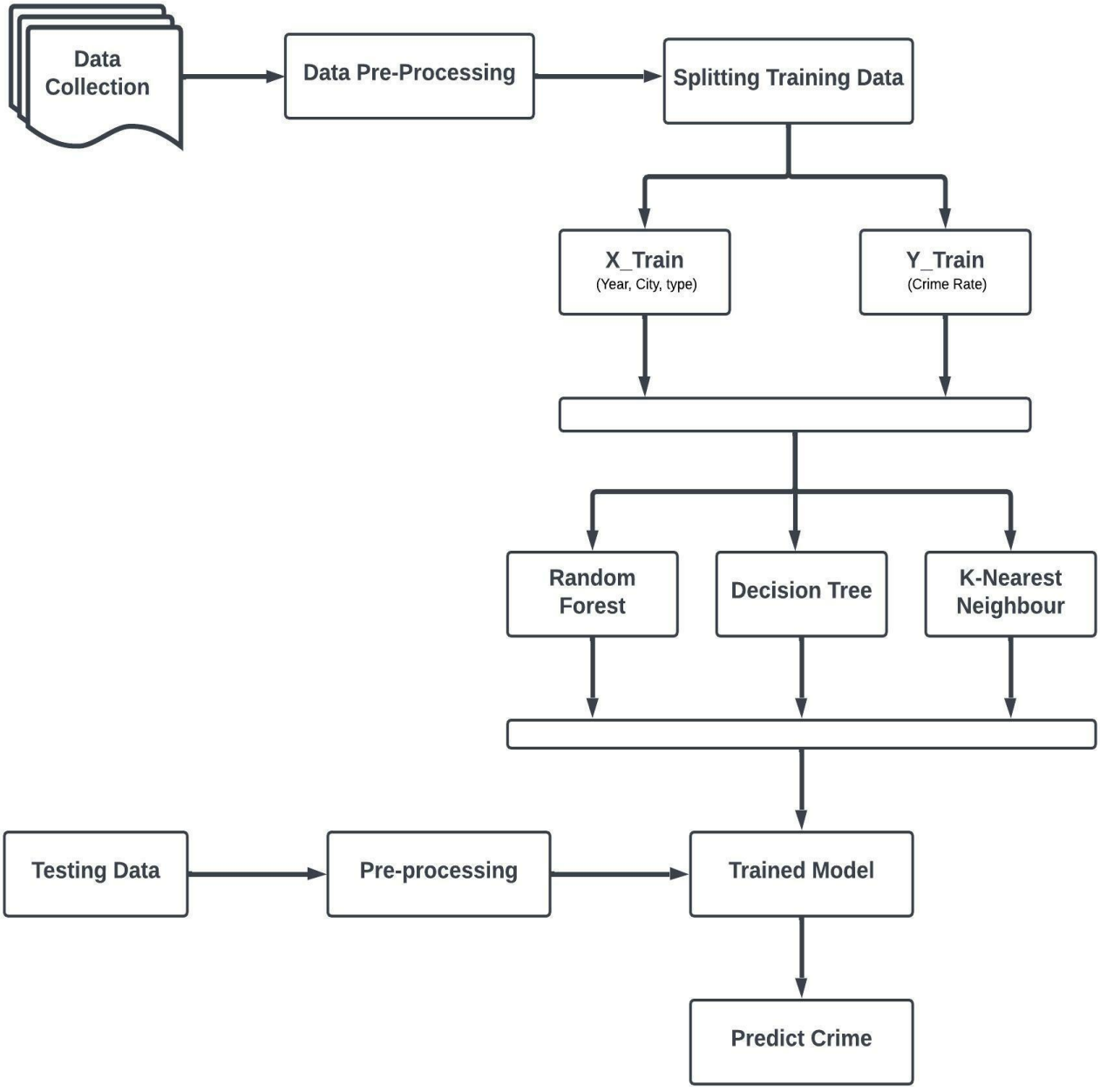


Figure 6.1 System Architecture

System Architecture Overview:

The system architecture for the crime prediction system is designed to analyze historical crime data and predict future crime rates using machine learning techniques. The process follows a structured pipeline that includes data preparation, model training, and prediction.

Data Collection:

Data is gathered from reliable sources, including crime records by year, city, and crime type. This is the foundation of the system as the quality and quantity of data directly impact the model's performance.

Data Pre-Processing:

Raw data often contains noise, missing values, or inconsistencies. This step involves cleaning the data, handling null values, and converting categorical values into numerical formats suitable for machine learning.

Splitting Training Data:

The pre-processed data is split into:

X_Train: Features like year, city, and crime type.

Y_Train: The target output, i.e., the crime rate. This separation allows supervised learning models to learn the relationship between inputs and outputs.

Model Training:

Three different algorithms are trained with the input data:

Random Forest:

Random Forest is a powerful ensemble machine learning algorithm used for both classification and regression tasks. It works by building multiple decision trees during training and combining their outputs to improve accuracy and prevent overfitting. Random Forest is known for its high accuracy, robustness, and ability to handle large datasets with many features.

Decision Tree:

A Decision Tree is a machine learning algorithm that models decisions using a tree-like structure. Each internal node represents a test on a feature, each branch represents an outcome of that test, and each leaf node gives the final prediction. It is easy to understand and interpret, and is used for both classification and regression tasks.

K-Nearest Neighbour (KNN):

K-Nearest Neighbour (KNN) is a simple, instance-based machine learning algorithm used for classification and regression. It works by finding the ‘k’ closest data points (neighbors) to a given input and making predictions based on their values. For classification, it assigns the most common class among the neighbors; for regression, it averages their values. KNN is easy to implement and works well with smaller datasets.

Model Evaluation and Prediction:

Testing Data:

After model training, a separate dataset (not used in training) is used to evaluate the model's performance. This ensures the model can generalize well to new, unseen data.

Pre-processing Testing Data:

Like training data, the testing data must also be pre-processed in the same manner (normalization, encoding, etc.) to maintain consistency with the trained model.

Trained Model:

The machine learning models, once trained and evaluated, are saved and ready to make predictions. The model with the best accuracy may be selected for deployment.

Predict Crime:

The trained model takes new input data (e.g., upcoming year, city, and crime type) and predicts the expected crime rate. These predictions can be used by law enforcement or authorities for better resource allocation and crime prevention strategies.

7. SYSTEM STUDY

7.1 FEASIBILITY STUDY

Three key considerations involved in the feasibility analysis are

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

7.2 TECHNICAL FEASIBILITY

The project uses Python and widely available machine learning libraries like Scikit-learn, making it technically feasible on standard computing devices without the need for expensive hardware or software.

7.3 ECONOMIC FEASIBILITY

Since the project uses open-source tools and publicly available datasets, the cost of implementation is minimal, making it economically viable.

7.4 OPERATIONAL FEASIBILITY

The system is user-friendly and can be operated by analysts with basic knowledge of Python and data analysis. It provides clear visualizations and predictions that aid decision-making.

8. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.1TYPES OF TESTS

8.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.1.2 Integration testing

After unit testing, the different modules (data loading, preprocessing, model

training, prediction, and visualization) were integrated and tested together to confirm smooth data flow and interaction between components. Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.1.3 Performance Testing:

Performance testing was conducted to evaluate the accuracy and effectiveness of various machine learning models in predicting crime rates based on historical data. The models tested included Random Forest Regressor, Decision Tree Regressor, K-Nearest Neighbour (KNN), and Support Vector Regressor (SVR). The dataset was split into training and testing sets, and each model was trained using the training data. Predictions were then made on the test data, and the results were evaluated using the R^2 Score, a standard metric for regression that measures how well the predicted values align with the actual values.

8.1.4 Model Testing:

Model testing was carried out to assess and validate the predictive capability of different machine learning algorithms on unseen data. After training models such as Random Forest Regressor, Decision Tree Regressor, K-Nearest Neighbour (KNN), and Support Vector Regressor (SVR) on the training dataset, each model was tested using a separate test dataset to evaluate how well it generalized to new inputs. This testing phase was critical in identifying the most reliable model for deployment in the application interface, ensuring robust and accurate crime rate forecasts.

9. IMPLEMENTATION

- **K nearest Neighbors**
  - This algorithm is used as one of the models to predict crime rates based on historical data. It classifies data points by measuring distances and assigning the label most common among the k closest neighbors.

  Syntax:

```
knn = KNeighborsRegressor(n_neighbors=3)
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)
print("KNN R2 Score:", r2_score(y_test, y_pred_knn))
```

- **Support Vector Machine**
  - This algorithm is used to predict crime rates by finding the optimal hyperlane that separates data points of different crime categories.

  Syntax:

```
svr = SVR(kernel='rbf')
svr.fit(X_train, y_train)
y_pred_svr = svr.predict(X_test)
print("SVR R2 Score:", r2_score(y_test, y_pred_svr))
```

- **Random Forest Regressor**
  - It is used to predict crime rates by building multiple decision trees and Averaging their outputs for accurate results.

Syntax:

```
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print("Random Forest R2 Score:", r2_score(y_test, y_pred_rf))
```

- **Decision Tree Regressor**
  - It splits the data into branches based on feature values to make accurate and interpretable predictions.

Syntax:

```
dt = DecisionTreeRegressor(random_state=42)
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)
print("Decision Tree R2 Score:", r2_score(y_test, y_pred_dt))
```

- **Neural Network MLP Regressor**
  - It captures complex patterns in the data for more accurate and non-linear predictions.

Syntax:

```
Mlp = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=500,
activation='relu', solver='adam', random_state=42)
mlp.fit(X_train, y_train)
y_pred_mlp = mlp.predict(X_test)
score = r2_score(y_test, y_pred_mlp)
print("MLP Regressor R2 Score:", score)
```

10. MODULE DESCRIPTION

- This research work attempts to implement various classification algorithms.
- To perform disease prediction.
  - Data Collection
  - Data Preprocessing
  - Exploratory Data Analysis Module
  - Model Training Module
  - Model Evaluation Module
  - Prediction Module
  - Visualization Module

Data Collection

Responsible for gathering historical crime data from sources such as public datasets or APIs. Ensures data is stored in a structured format (CSV or database) for further processing.

Data Preprocessing Module

Handles cleaning of the raw data by removing null or inconsistent values. Encodes categorical variables (e.g., crime type, location) into numerical format for machine learning algorithms. Scales numerical features to normalize the data. Splits the dataset into training and testing sets.

Exploratory Data Analysis (EDA) Module

Analyzes data trends and patterns using statistical and visualization techniques. Generates graphs such as bar charts, heatmaps, and line plots to identify crime

hotspots and temporal trends. Helps in feature selection and understanding relationships between variables.

Model Training Module

Implements machine learning algorithms such as Linear Regression and Random Forest Regressor. Trains models on the prepared training data. Optimizes model parameters for better accuracy.

Model Evaluation Module

Tests the trained models on the testing dataset. Calculates performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2).Compares different models to select the best-performing one.

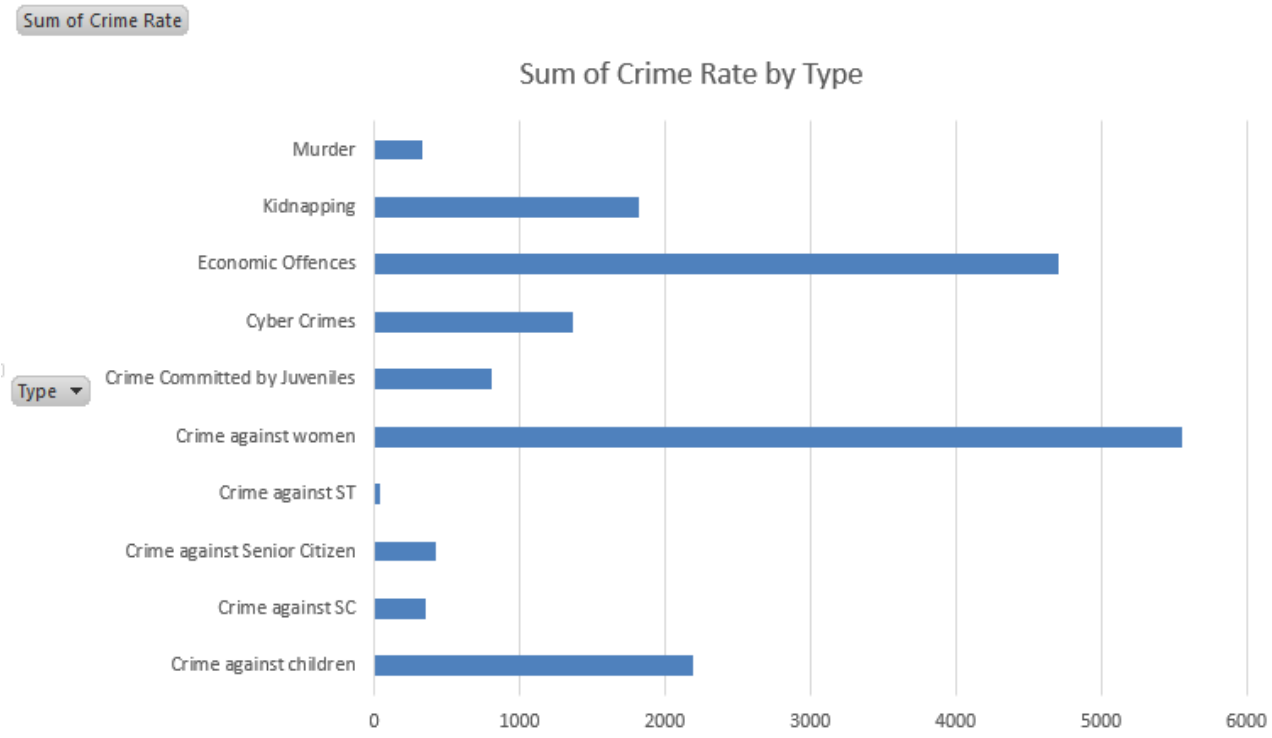
Prediction Module

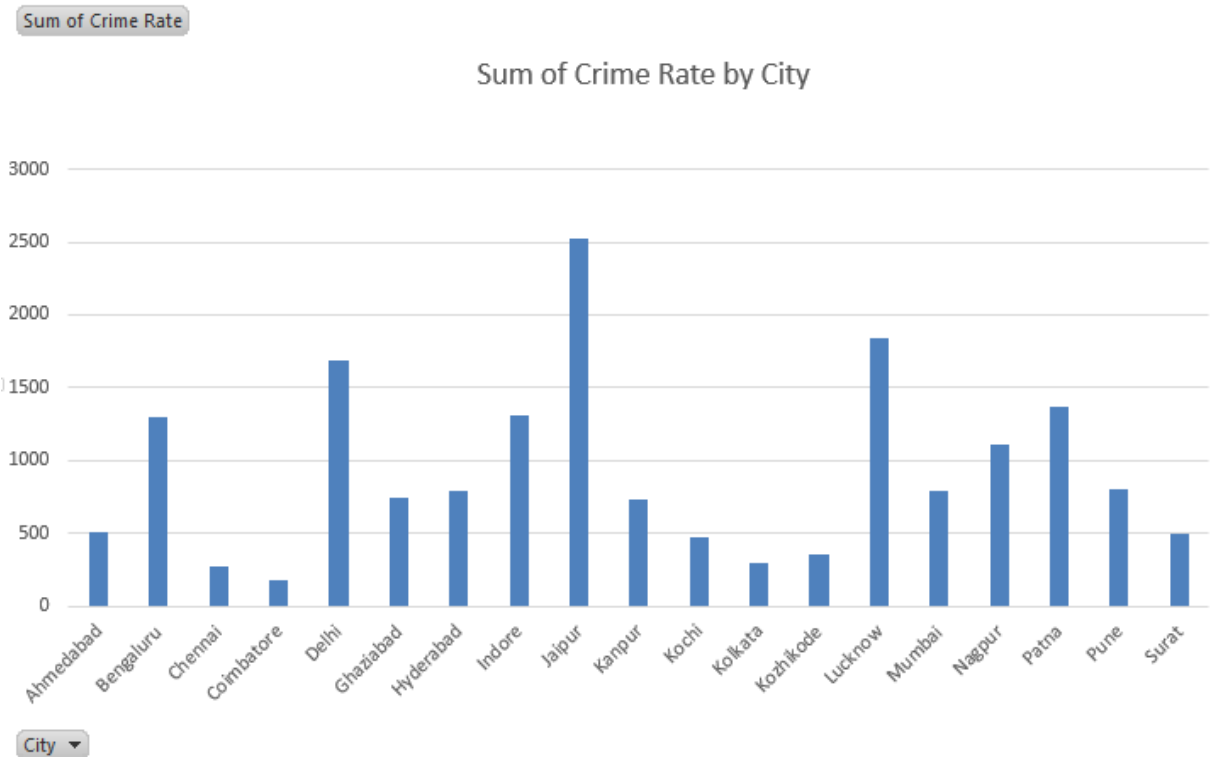
Uses the selected machine learning model to predict future crime rates based on input features. Provides predicted crime values for given locations and time frames.

Visualization Module

Displays the prediction results and crime trends through graphical representations. Enables users to intuitively understand crime patterns and predicted rates.

11. Comparison chart



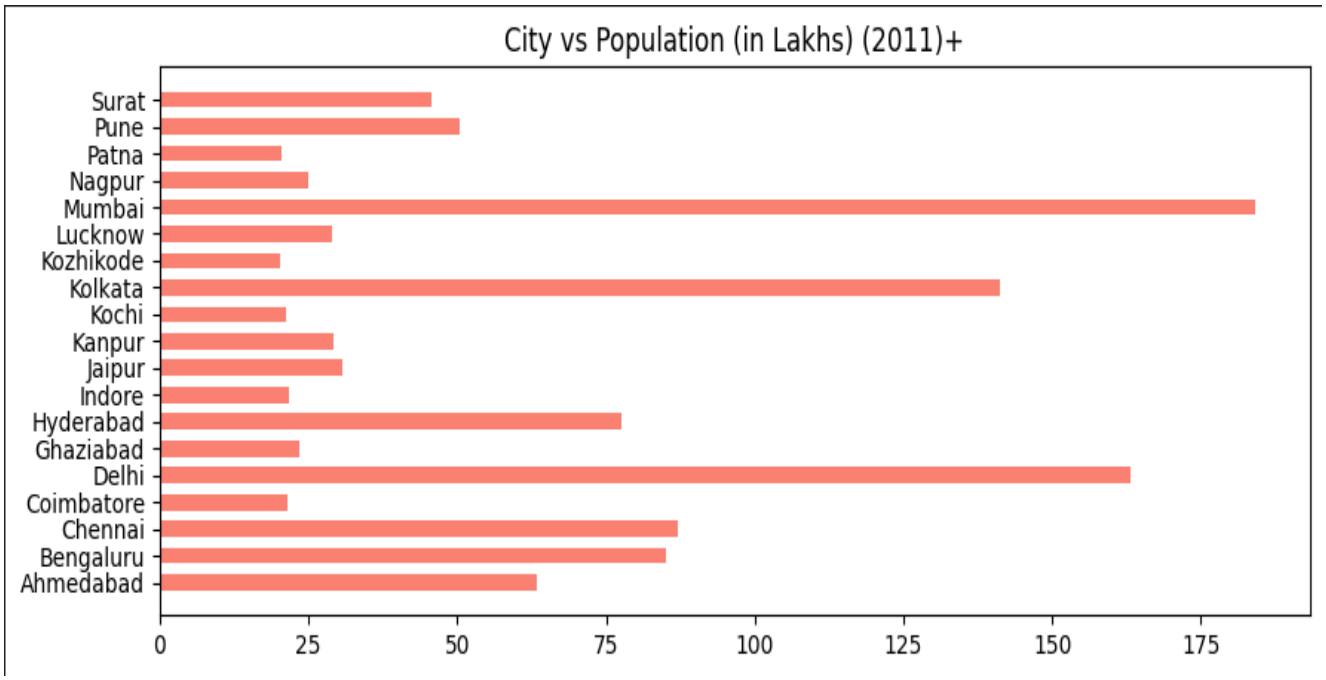


12. Output Screenshot

Data Collection:

	Year	City	Population (in Lakhs) (2011)+	Murder	Kidnapping	Crime against women	Crime against children	Crime Committed by Juveniles	Crime against Senior Citizen	Crime against SC	Crime against ST	Economic Offences	Cyber Crimes
0	2014	Ahmedabad	63.5	82	367	1371	437	215	68	66	6	399	32
1	2015	Ahmedabad	63.5	94	332	1067	609	157	17	60	9	378	28
2	2016	Ahmedabad	63.5	103	376	1126	481	258	362	96	10	479	77
3	2017	Ahmedabad	63.5	90	263	1405	600	405	534	119	6	608	112
4	2018	Ahmedabad	63.5	98	277	1416	733	352	733	145	9	842	212
...	...	...	...	...	...	...	...	...	...	...	...	...	...
147	2017	Surat	45.8	89	332	559	526	436	131	32	10	719	105
148	2018	Surat	45.8	108	373	712	1075	409	161	29	13	829	155
149	2019	Surat	45.8	97	358	1015	770	516	232	34	19	804	228
150	2020	Surat	45.8	116	163	633	419	298	69	20	12	401	204
151	2021	Surat	45.8	121	270	622	479	355	66	22	19	663	296

Data Visualization:



Data Preprocessing:

	Year	City	Population (in Lakhs) (2011)+	Number Of Cases	Type
0	2014	Ahmedabad	63.5	82	Murder
1	2015	Ahmedabad	63.5	94	Murder
2	2016	Ahmedabad	63.5	103	Murder
3	2017	Ahmedabad	63.5	90	Murder
4	2018	Ahmedabad	63.5	98	Murder
...	...	...	...	...	...
147	2017	Surat	45.8	105	Cyber Crimes
148	2018	Surat	45.8	155	Cyber Crimes
149	2019	Surat	45.8	228	Cyber Crimes
150	2020	Surat	45.8	204	Cyber Crimes
151	2021	Surat	45.8	296	Cyber Crimes

Splitting of dataset for training:

```
[26]
y = new_dataset['Crime Rate'].values
y

array([1.29133858, 1.48031496, 1.62204724, ..., 4.97816594, 4.45414847,
       6.4628821 ])

[27]
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=50)

[28]
x_train

array([[2.018e+03, 8.000e+00, 3.070e+01, 1.000e+00],
       [2.015e+03, 1.200e+01, 2.030e+01, 3.000e+00],
       [2.014e+03, 5.000e+00, 2.360e+01, 6.000e+00],
       ...,
       [2.015e+03, 1.200e+01, 2.030e+01, 7.000e+00],
       [2.019e+03, 1.300e+01, 2.900e+01, 9.000e+00],
       [2.014e+03, 1.700e+01, 5.050e+01, 6.000e+00]])

[29]
y_train

array([ 5.86319218,  0.98522167,  2.58474576, ..., 12.95566502,
        2.5862069 ,  2.77227723])
```

Model creation:

Support vector machine:

```
Mean Absolute Error: 10.320485537023602
Mean Squared Error: 371.7907414958839
R2 score: -0.17886853639175593
```

Nearest neighbor:

```
Mean Absolute Error: 6.743751501712896
Mean Squared Error: 143.15382657114714
R2 score: 0.5460899824191134
```

Decision tree regressor:

```
Mean Absolute Error: 2.886686146144012
Mean Squared Error: 34.87170291005993
R2 score: 0.8894293246634664
```

Random forest regressor:

```
Mean Absolute Error: 2.487973049950136
Mean Squared Error: 21.362971441526998
R2 score: 0.9322626088672231
```

Neural networks MLP regressor:

```
Mean Absolute Error: 12.424838196292534
Mean Squared Error: 307.55062475298035
R2 score: 0.024823067378803798
```

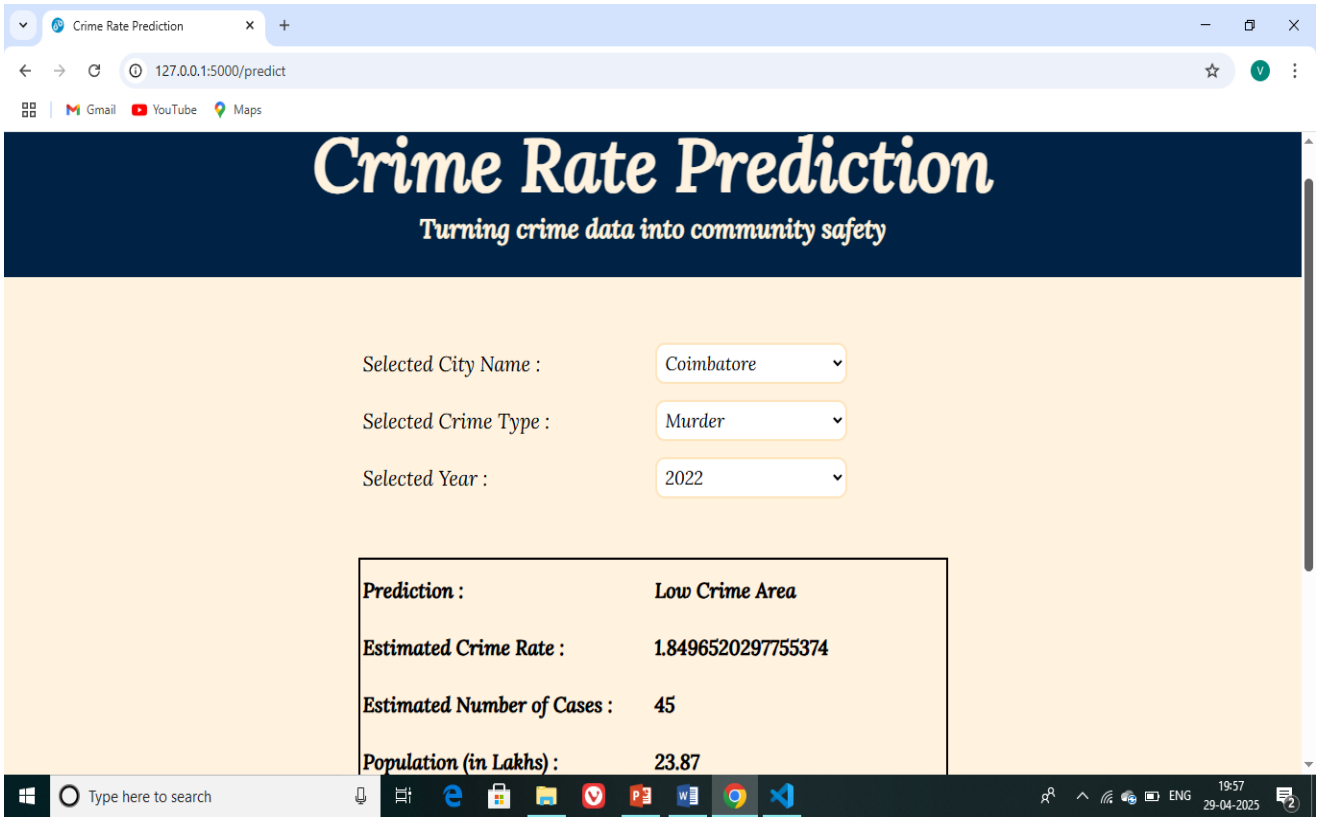
Saving the model:

```
#saving the model as .pkl file
pkl_filename = "model.pkl"
with open(pkl_filename, 'wb') as file:
    pickle.dump(model4, file)
```

```
#checking the saved model accuracy
with open(pkl_filename, 'rb') as file:
    pickle_model = pickle.load(file)
score = pickle_model.score(x_test, y_test)
print(score)
```

```
0.9322626088672231
```

Final Output:



13. CONCLUSION & FUTURE ENHANCEMENT

13.1 CONCLUSION

Crime rate prediction has become an important tool for law enforcement agencies to help them focus their resources in high-crime areas. With the help of sophisticated algorithms and data analysis, law enforcement agencies can predict when and where crimes are likely to occur. By focusing their resources in the right areas, police officers can help reduce the overall crime rate in a community. Predictive policing has already proven to be an effective tool in reducing crime rates in many areas, and it looks like it will continue to be a key tool in the future. As a result of machine learning technology, finding relationships and patterns between various data has become easier. The training data has been cleaned and transformed to create a machine learning model using the concept of machine learning. The model predicts the crime rate with an accuracy of 93.20%. The model prediction of crime rate and data visualization helps in analysis of data set and prediction of crimes. Many graphs are created to found interesting statistics that helped in understanding different crime datasets that can be used in implementing the factors that can help in keeping society safe.

13.2 FUTURE ENHANCEMENT

- **Web based platform:** Develop an interactive website for real-time crime prediction and visualization.
- **Integration of real time data:** Incorporate live inputs from social media, emergency calls and surveillance sytems.
- **Geospatial crime mapping:** Add location based heatmaps for identifying crime hotspots using GIS.

14. APPENDIX

Result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Crime Rate Prediction</title>
    <link rel="icon" href="static/images/favicon.png" type="image/png">
    <link rel="stylesheet" href="static/styles.css">
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Lora:ital,wght@0,400;0,500;1,400;1
,500&display=swap');
    </style>
</head>
<body>
    <header>
        <h1 class = "title">Crime Rate Predictor</h1>
        <h2 class = "tagline">Unlock safety: Reduce crime rate together</h2>
    </header>
    <section class="prediction">
        <table class="table2">
            <tr class="r1">
```

```
    <td class="label">
        <label>Selected City Name :</label>
    </td>
    <td class="sel">
        <select>
            <option value="selected">{{city_name}}</option>
    </td>
</tr>
<tr class="r1">
    <td class="label">
        <label>Selected Crime Type :</label>
    </td>
    <td class="sel">
        <select >
            <option value="selected">{{crime_type}}</option>
    </td>
</tr>
<tr class="r1">
    <td class="label">
        <label>Selected Year :</label>
    </td>
    <td class="sel">
        <select>
            <option value="selected">{{year}}</option>
    </td>
```

```
        </tr>
    </table>
    <table class="table3">
        <tr class="r2"">
            <td class="result">Prediction :</td>
            <td class="result">{{crime_status}}</td>
        </tr>
        <tr class="r2">
            <td class="result">Estimated Crime Rate :</td>
            <td class="result">{{crime_rate}}</td>
        </tr>
        <tr class="r2">
            <td class="result">Estimated Number of Cases :</td>
            <td class="result">{{cases}}</td>
        </tr>
        <tr class="r2">
            <td class="result">Population (in Lakhs) :</td>
            <td class="result">{{population}}</td>
        </tr>
    </table>
    <div class="button2" >
        <a href="http://127.0.0.1:5000">
            <input type="submit" class="btn id="check_again"" value="Let's Check
Again" >
        </a>
```

```
        </div>
    </section>
    <footer>
        <div class="footer">
            <div id="developer">
                <h3 >Developed By:   Abhay Rautela</h3>
            </div>
            <div id="linkedin">
                <a href="https://www.linkedin.com/in/tushar-sharma-2001/"
target="_blank">
                    <img src="static/images/linkedin.png" alt="linkedIn" width="35" >
                </a>
            </div>
            <div id="github">
                <a href="https://github.com/Tushar0804" target="_blank">
                    <img src="static/images/github.png" alt="GitHub" width="35">
                </a>
            </div>
        </div>
    </footer>
</body>
</html>
```

Application code (app.py)

```
from flask import Flask, request, render_template
import pickle
import math
import os

# Load the trained model
model_path = os.path.join('Model', 'model.pkl')
with open(model_path, 'rb') as file:
    model = pickle.load(file)
app = Flask(_name_)
@app.route('/')
def index():
    return render_template("index.html")

@app.route('/predict', methods=['POST'])
def predict_result():
    city_names = {
        '0': 'Ahmedabad', '1': 'Bengaluru', '2': 'Chennai', '3': 'Coimbatore',
        '4': 'Delhi', '5': 'Ghaziabad', '6': 'Hyderabad', '7': 'Indore',
        '8': 'Jaipur', '9': 'Kanpur', '10': 'Kochi', '11': 'Kolkata',
        '12': 'Kozhikode', '13': 'Lucknow', '14': 'Mumbai', '15': 'Nagpur',
        '16': 'Patna', '17': 'Pune', '18': 'Surat'
    }
```

```
    crimes_names = {
        '0': 'Crime Committed by Juveniles', '1': 'Crime against SC', '2': 'Crime against
ST',
        '3': 'Crime against Senior Citizen', '4': 'Crime against children',
        '5': 'Crime against women', '6': 'Cyber Crimes', '7': 'Economic Offences',
        '8': 'Kidnapping', '9': 'Murder'
    }

    population = {
        '0': 63.50, '1': 85.00, '2': 87.00, '3': 21.50, '4': 163.10, '5': 23.60,
        '6': 77.50, '7': 21.70, '8': 30.70, '9': 29.20, '10': 21.20, '11': 141.10,
        '12': 20.30, '13': 29.00, '14': 184.10, '15': 25.00, '16': 20.50, '17': 50.50, '18':
45.80
    }

    city_code = request.form["city"]
    crime_code = request.form['crime']
    year = int(request.form['year'])
    pop = population[city_code]
    year_diff = year - 2011
    pop += 0.01 * year_diff * pop

    input_features = [[year, int(city_code), pop, int(crime_code)]]
    crime_rate = model.predict(input_features)[0]
```

```
    city_name = city_names[city_code]
    crime_type = crimes_names[crime_code]

    if crime_rate <= 1:
        crime_status = "Very Low Crime Area"
    elif crime_rate <= 5:
        crime_status = "Low Crime Area"
    elif crime_rate <= 15:
        crime_status = "High Crime Area"
    else:
        crime_status = "Very High Crime Area"

    cases = math.ceil(crime_rate * pop)
    return render_template('result.html',
                           city_name=city_name,
                           crime_type=crime_type,
                           year=year,
                           crime_status=crime_status,
                           crime_rate=crime_rate,
                           cases=cases,
                           population=round(pop, 2))

if _name_ == '_main_':
    app.run(debug=False)
```

15. REFERENCES

1. M. Alkaff, N. F. Mustamin, and G. A. A. Firdaus, “Prediction of Crime Rate in Banjarmasin City Using RNN-GRU Model”, Int J Intell Syst Appl Eng, vol. 10, no. 3, pp. 01–09, Sep. 2022.
2. W. Safat, S. Asghar and S. A. Gillani, “Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques”, in IEEE Access, vol. 9, pp.
3. Mahmud, S., Nuha, M., Sattar, A. (2021). “Crime Rate Prediction Using Machine Learning and Data Mining”. In: Borah, S., Pradhan, R., Dey, N., Gupta, P. (eds) Soft Computing Techniques and Applications. Advances in Intelligent Systems and Computing, vol 1248. Springer, Singapore. https://doi.org/10.1007/978-981-15-7394-1_5
4. Gaurav Hajela, Meenu Chawla, Akhtar Rasool, “A Clustering Based Hotspot Identification Approach for Crime Prediction”, Procedia Computer Science, Volume 167, 2020, Pages 1462-1470, ISSN 1877-0509. https://doi.org/10.1016/j.procs.2020.03.357.
5. Fatehkia, Masoomali & O’Brien, Dan & Weber, Ingmar. (2019). Correlated impulses: Using Facebook interests to improve predictions of crime rates in urban areas. PLOS ONE. 14. e0211350. 10.1371/journal.pone.0211350.
6. Ms. Vrushali Pednekar, Ms. Trupti Mahale, Ms. Pratiksha Gadhave, Prof. Arti Gore. 2018. “Crime Rate Prediction Using KNN”. International Journal on Recent and Innovation Trends in Computing and Communication 6(1) : 124 - https://doi.org/10.17762/ijritcc.v6i1.1392.