A PROJECT REPORT ON

CRIME DATA ANALYSIS USING MACHINE LEARNING

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

For Partial Fulfilment of Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

CSE-ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Submitted By

V. Gowri Ganesh	**20X41A4257**
K. Durga Sai	**20X41A4229**
M. Venkata Prem Shankar	**20X41A4232**
G. Sarvan	**20X41A4219**

Under the esteemed guidance of

MR. J. BALA SRINIVAS RAO

Assistant Professor, Department of CSE-Artificial Intelligence and Machine Learning

DEPARTMENT OF CSE-ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

S.R.K INSTITUTE OF TECHNOLOGY

(Approved AICET, New Delhi &Affiliated To JNTUK, Kakinada)

(An Iso 9001:2015 Certified Institution & Accredited By NAAC With ‘A’ Grade)

Enikepadu, Vijayawada – 521108.

APRIL 2024

S.R.K INSTITUTE OF TECHNOLOGY

(Approved AICET, New Delhi &Affiliated To JNTUK, Kakinada)

(An Iso 9001:2015 Certified Institution & Accredited By NAAC With ‘A’ Grade)

Enikepadu, Vijayawada – 521108.

DEPARTMENT OF CSE- ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify that this project report entitled **“Crime Data Analysis using Machine Learning”** is the bonafide work of **V. Gowri Ganesh (20X41A4257), K. Durga Sai (20X41A4229), M. Venkata Prem Shankar (20X41A4232), G. Sarvan (20X41A4219) in** partial fulfillment of the requirements for the award of the graduate degree in **BACHELOR OF TECHNOLOGY** during the academic year 2023-2024. This Work has carried out under our supervision and guidance.

(MR.J. BALA SRINIVAS RAO)	**(Dr. D.ANUSHA)**
Signature of the Guide	**Signature of the HOD**

Signature of the External Examiner

DECLARATION

We V. Gowri Ganesh, K. Durga Sai, M. Venkata Prem Shankar, G. Sarvan hereby declare that the project report entitled "**Crime Data Analysis using Machine Learning**" is an original work done in the Department CSE- Artificial Intelligence And Machine Learning, SRK Institute of Technology, Enikepadu, Vijayawada, during the academic year 2023-2024, in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering. We assure you that this project is not submitted to any other College or University.

PROJECT ASSOCIATES

Roll No	Name of the Student	Signature
20X41A4257	V. Gowri Ganesh	
20X41A4232	M. Venkata Prem Shankar	
20X41A4229	K. Durga Sai	
20X41A4219	G. Sarvan	

ACKNOWLEDGEMENT

Firstly, we would like to convey our heart full thanks to the Almighty for the blessings on us to carry out this project work without any disruption.

We are extremely thankful to **MR. J. BALA SRINIVAS RAO**, our guide throughout the project. We also thank his for most independence and freedom throughout the given to us during various phases of the project.

We are also thankful for our project coordinator **Dr. D. Anusha**, for her valuable guidance which helped us to bring this project successfully.

We are very much grateful to **Dr. D**. **Anusha** H.O.D of C.S.E - Artificial Intelligence and Machine Learning, for her valuable guidance which helped us to bring out this project successfully. Her wise approach made us learn the minute details of the subject. Her matured and patient guide paved away for completing our project with a sense of satisfaction and pleasure.

We are greatly thankful to our principal **Dr. M. Ekambaram Naidu** for his kind support and facilities provided at our campus which helped us to bring out this project successfully.

Finally, we would like to convey our heart full thanks to our Technical Staff, for their guidance and support in every step of this project. We convey our sincere thanks to all the faculty and friends who directly or indirectly helped us with the successful completion of this project.

V. Gowri Ganesh	**20X41A4257**
K. Durga Sai	**20X41A4229**
M. Venkata Prem Shankar	**20X41A4232**
G. Sarvan	**20X41A4219**

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

Criminal cases are rapidly increasing in our society day to day. These leading to backlog of pending cases. It is important to control the increasing crimes else it becomes tough to handle for Law Enforcement Agencies, they store the information of every crime after happening because there might be a chance of Pattern behind the occurrence of every crime so in order to control the crimes we are going to create a Machine Learning Model to predict the Crime pattern by training the model through K Nearest Neighbour (KNN) Algorithm to get more predictions accurately than existed Algorithms like Naïve Bayes, Decision Tree etc. Already the people worked on KNN algorithm to train model they used the dataset and get prediction rate of 75% but now we are working on same algorithm to get more than 85%. Also we used to work on this approach to reduce the code complexity. We would show the chance of occurrence of next two crime types based on our dataset. We collected the dataset from data.govern.in at the period of 2023 Jan - 2024 Jan.

KEYWORD:

- KNN
- Machine Learning
- Crime Prediction
- Data Visualization
- Accuracy
- Pattern Recognition

Chapter 1

INTRODUCTION

Crimes are harmful actions that lead to threats to human lives. Crimes might be Robbery, Murder, Rape, Women trafficking, etc.. As the population increases the rate of crime also increases day by day. The increasing cases lead to a backlog of pending cases to the police department, The crime activities have increased at a faster rate and it is the responsibility of the police department to control and reduce the crime activities. the department tries to solve the cases according to the evidence they got but in major cases, it is not as much possible to solve and decrease the crime rate as they think.

This analysis leads us to research the crimes to make them complex and free for solving the cases. The main thing here we are going to work on is predicting the occurance of the next crime. It might be helpful to the Law Enforcement agencies and police departments to control and be well aware of the respective situation.

It will only be possible by collecting the previous information. So we get the information stored in dataset format in which the dataset contains the relative features like crime type, place, time, arrest or not, victims, and whether the case is solved or not, etc. We could extract the dataset from official site data.govern.in . The prediction of the occurrence of crime can happen by working with a machine learning model and one optimal algorithm, here we are going to work with K Nearest Neighbour which is well-suited algorithm for both classification and regression and can also get good prediction accuracy. Visualization of occurrence of crimes are well-known thing to the normal people hence we could also implement the work with visual graphs.

1.1 Overview

Our project on crime data analysis features a user-friendly web interface comprising Home, Index, and Prediction pages. Upon accessing the Home page, users are seamlessly directed to the Index page. Here, they can upload their crime datasets and specify the target variable for analysis. Leveraging the K-Nearest Neighbors (KNN) algorithm implemented in app.py, our system predicts the occurrence of the next two crimes based on the uploaded dataset. The Prediction page not only displays the accuracy of the predictive model but also provides visualizations of prediction results through graphs. By integrating web-based functionality with advanced machine learning techniques, our project aims to offer a comprehensive platform for crime data analysis, empowering users, particularly law enforcement agencies, to make informed decisions and enhance community safety.

1.2 About the project

In this application we uses a time when there are lots of criminal cases happening in India, and the police have a hard time keeping up. But now, they have a new tool machine learning. This fancy technology helps them guess which crimes might happen next in certain areas. By looking at past crimes and information about the people who live there, like where they're from and how old they are, the computer can make pretty good guesses about what might happen next. This means the police can get ready ahead of time and try to stop crimes before they even happen. It's like having a crystal ball, but way cooler! And because of this, the police can do their job better, making our neighborhoods safer for everyone.

1.3 Purpose:

The purpose of this project is to help the police in India deal with lots of criminal cases by using machine learning. With this technology, they can predict which crimes might happen next in different areas. By knowing this, they can plan ahead and try to stop crimes before they happen, making our neighborhoods safer.

1.4 Scope:

The scope of this project is to use machine learning to predict future crimes in India. By analyzing past crime data and local information, the goal is to help law enforcement agencies anticipate where crimes might occur next. This will assist them in allocating resources effectively and taking preventive measures to enhance community safety.

Chapter 2

LITERATURE REVIEW

1.Mugdha Sharma, "Z-Crime: A Data Mining Tool for the Detection of Suspicious Criminal Activities based on the Decision Tree", International Conference on Data Mining and Intelligent Computing, pp. 1-6, 2014.

In her paper titled "Z-Crime: A Data Mining Tool for the Detection of Suspicious Criminal Activities based on the Decision Tree," Mugdha Sharma presents an innovative approach to combating criminal activities using data mining techniques. This research, presented at the International Conference on Data Mining and Intelligent Computing in 2014, introduces a tool called Z-Crime, which leverages decision tree algorithms to detect suspicious criminal activities. By analyzing various factors and patterns within crime data, Z-Crime aims to identify potential criminal behaviors and activities before they occur.

The tool operates by analyzing various parameters related to criminal activities, such as location, time, and type of offense, to generate decision trees that can predict the likelihood of suspicious behavior. These decision trees serve as a guide for law enforcement personnel, enabling them to prioritize their efforts and allocate resources effectively in areas where criminal activities are more likely to occur. Through the implementation of Z-Crime, law enforcement agencies can enhance their crime detection capabilities and ultimately contribute to maintaining public safety.

Sharma's work underscores the importance of utilizing advanced data mining methodologies in law enforcement to proactively address criminal activities. By employing decision tree algorithms, Z-Crime facilitates the classification and prediction of suspicious behaviors, enabling law enforcement agencies to take preemptive measures. This research contributes significantly to the field of crime detection and prevention, offering a promising tool for enhancing public safety and security.

2. HARVARD University—"Minority Report" a Reality? The NYPD's big data approach to predicting crime. Available at https://digital.hbs.edu/platform-rctom/submission/minority-report-a-reality-the-nypds-big-data-approach-to-predicting-crime/. Accessed 08 Nov 2019.

In the article "Minority Report a Reality? The NYPD's big data approach to predicting crime" from Harvard University, the focus is on the New York Police Department's (NYPD) use of big data to predict and prevent crimes. Drawing parallels to the science fiction movie "Minority Report," the article discusses how the NYPD utilizes advanced analytics and technology to forecast criminal activities before they happen. By analyzing vast amounts of data, including past crime records, demographics, and environmental factors, the NYPD aims to identify patterns and trends that can help in proactive policing strategies. The article explores the ethical implications and concerns surrounding the use of such predictive policing methods, highlighting the balance between public safety and individual privacy rights. Overall, it provides insights into the intersection of technology, law enforcement, and societal impact in the realm of crime prediction.

3. Bolger MA, Colin Bolger P (2019) Predicting fear of crime: results from a community survey of a small city. Am J Crim Just. 44(2):334–351.

In the study conducted by Bolger and Bolger in 2019, titled "Predicting fear of crime: results from a community survey of a small city," the authors investigate the factors influencing the fear of crime within a small city community. Published in the American Journal of Criminal Justice, the research delves into understanding the psychological aspects and predictors associated with fear of crime among residents. Through a comprehensive community survey, the study analyzes various socio-demographic factors, perceived safety, and neighborhood characteristics to determine their impact on individuals' fear of crime. The findings shed light on the nuanced dynamics shaping fear perception and contribute to the existing literature on crime psychology and community safety. This research provides valuable insights for policymakers and law enforcement agencies in devising strategies to address and alleviate the fear of crime among urban residents.

4. Mittal M, Goyal LM, Sethi JK, Hemanth DJ (2019) Monitoring the impact of economic crisis on crime in India using machine learning. Comput Econ 53(4):1467–1485

In their 2019 study titled "Monitoring the impact of economic crisis on crime in India using machine learning," Mittal, Goyal, Sethi, and Hemanth explore the relationship between economic downturns and crime rates in India. Published in the journal Computational Economics, the research employs machine learning techniques to analyze the effects of economic crises on various types of crime. By examining crime data alongside economic indicators, the study aims to uncover patterns and correlations that can help understand how economic downturns influence criminal behavior. The findings provide valuable insights into the socioeconomic factors contributing to crime trends in India and offer implications for policy interventions aimed at mitigating crime during periods of economic instability. This research contributes to the interdisciplinary field of computational economics and crime analysis, emphasizing the potential of machine learning in understanding complex societal phenomena.

5. Vural MS, Gok M (2017) Criminal prediction using Naive Bayes theory. Neur Comput Appl 28(9):2581–2592

In their 2017 study titled "Criminal prediction using Naive Bayes theory," Vural and Gok investigate the application of Naive Bayes theory in predicting criminal activities. Published in the journal Neural Computing and Applications, the research explores how this probabilistic algorithm can be utilized to forecast criminal behavior based on relevant data variables. By analyzing patterns and relationships within crime data, the study aims to develop a predictive model that can assist law enforcement agencies in identifying potential criminal activities before they occur. The findings highlight the efficacy of Naive Bayes theory in crime prediction and its potential utility in enhancing proactive policing strategies. This research contributes to the growing body of literature on machine learning approaches to crime prevention and underscores the importance of leveraging advanced analytics in law enforcement efforts.

6. Chen, H., Chungcdvsfdf, W., Xu, J. J., Wangsac, G., Qin, Y., & Chauascas, M. (2004). Crime datamining: a general framework and some examples. computer, 37(4), 50-56.

In their 2004 paper titled "Crime data mining: a general framework and some examples," Chen, Chung, Xu, Wang, Qin, and Chauascas present a comprehensive framework for mining crime data. Published in the journal Computer, the research introduces a systematic approach to analyzing crime data using data mining techniques. The framework encompasses various stages, including data preprocessing, pattern discovery, and evaluation, aimed at extracting meaningful insights from large-scale crime datasets. The authors provide illustrative examples to demonstrate the application of their framework in real-world scenarios, showcasing its effectiveness in identifying crime patterns, trends, and anomalies. This paper contributes to the field of crime analytics by offering a structured methodology for leveraging data mining to enhance crime prevention and law enforcement strategies.

Chapter 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Many researchers have gone through this problem regarding the criminal cases being unsolved for a long period. They proposed different crime prediction algorithms. In all these models the accuracy will surely vary depending on the data set and the features or attributes we select during data pre-processing. In Crime prediction done on the Mississippi crime data set where models like linear regression and Decision stump model are used gave a result of 75%, 69% and 67% respectively [1]. Although these accuracies of the predictions may vary accordingly because it is discovered that many machine learning algorithms are implemented on data sets consisting of different places having distinctive features, so predictions are changing in all ca algorithm which is a supervised learning algorithm which is used for classification. Mainly used for text classification based on the training dataset. Naïve Bayes algorithm assumes all features were independent to each other.

It depends on the conditional probability.

Formula :

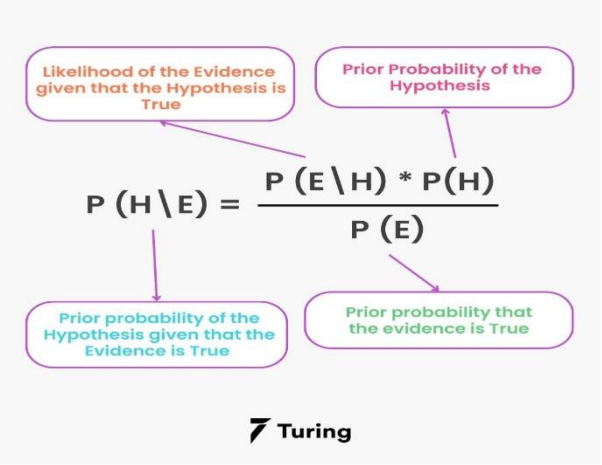

Likelihood of the Evidence given that the Hypothesis is True
Prior Probability of the Hypothesis
P (H\E) = P (E\H) * P(H) / P (E)
Prior probability of the Hypothesis given that the Evidence is True
Prior probability that the evidence is True
Turing

Fig: 3..1 Naïve Bayes formula.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

- Dependence on Data Quality
- Bias in data
- Limited Generalizability
- Sensitivity to Feature Selection

3.2 PROPOSED SYSTEM

The proposed system algorithm utilizes the K-Nearest Neighbors (KNN) approach for crime data analysis, aiming to provide actionable insights for proactive crime prevention strategies. KNN is a supervised machine learning algorithm commonly used for classification tasks. In the context of crime prediction, KNN works by identifying the similarity between a new data point (crime incident) and its nearest neighbors in the dataset. Based on the majority class among the nearest neighbors, KNN assigns a label to the new data point, predicting the type of crime it is likely to be.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

- Simple Implementation
- No Assumptions about Data Distribution
- Adaptability To Local Patterns
- Suitability for Multiclass Classification
- Robustnesss to Outliers
- Interpretability

3.2.2 METHODOLOGY

The methodology for our project involves several key steps. First, we will preprocess the dataset obtained from data.govern.in, ensuring it is cleaned and formatted properly for training. Then, we will implement the K Nearest Neighbors (KNN) algorithm to train our machine learning model on this dataset, striving to achieve a prediction rate of over 85%. To reduce code complexity, we will streamline the implementation of the KNN algorithm. Our goal is to accurately predict crime patterns by leveraging the insights gained from the dataset, ultimately enabling us to anticipate the occurrence of the next two crime types with a high degree of accuracy.

ALGORITHM STEPS:

1.Load the Data: Begin by loading the dataset containing information about past crimes. This dataset will typically include features such as the type of crime, location, time of occurrence, etc.

2.Preprocess the Data: Before applying the KNN algorithm, it's crucial to preprocess the data. This involves handling missing values, encoding categorical variables, scaling numerical features, and splitting the dataset into training and testing sets.

3.Choose the Value of K: Select the number of nearest neighbors, K, to consider when making predictions. This value can significantly impact the performance of the algorithm, so it may require experimentation and tuning.

4.Calculate Distances: For each data point in the test set, calculate its distance to all data points in the training set. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance.

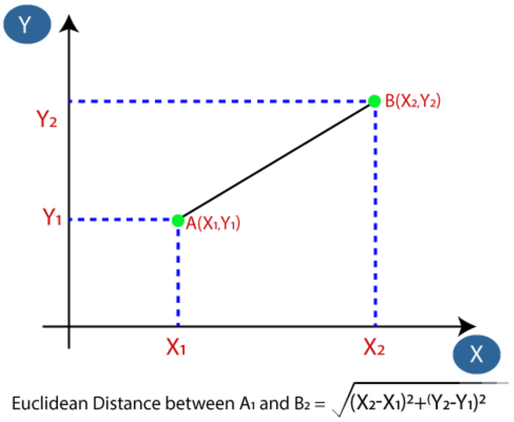
5.Find K Nearest Neighbors: Identify the K data points in the training set that are closest (most similar) to the data point being evaluated. This is typically done by sorting the distances calculated in the previous step and selecting the K nearest neighbors.

6.Majority Vote: Determine the class labels of the K nearest neighbors. For classification tasks, this can be done by counting the occurrences of each class among the neighbors and selecting the class with the highest frequency (mode).

7.Make Predictions: Assign the most common class label among the K nearest neighbors as the predicted class for the test data point. For regression tasks, the predicted value can be the average (or another aggregation) of the target variable for the K nearest neighbors.

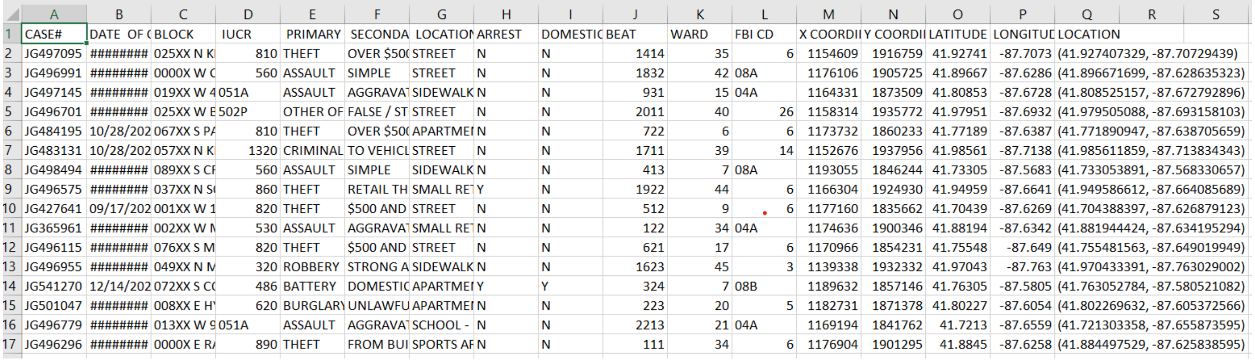
8.Evaluate Model Performance: Finally, assess the performance of the KNN model using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, or mean squared error (for regression). This step helps determine how well the model generalizes to unseen data and whether adjustments to the algorithm or its parameters are necessary.

FORMULA:


Y
X
Y_2
Y_1
X_1
X_2
$B(X_2,Y_2)$
$A(X_1,Y_1)$
Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

- FIG:3.2 Representation of KNN formula

3.2.3 DATASET DETAILS:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	CASE#	DATE OF (	BLOCK	IUCR	PRIMARY	SECONDA	LOCATIOI	ARREST	DOMESTIC	BEAT	WARD	FBI CD	X COORDII	Y COORDII	LATITUDE	LONGITUD	LOCATION		
2	JG497095	########	025XX N KI	810	THEFT	OVER $50(	STREET	N	N	1414	35	6	1154609	1916759	41.92741	-87.7073	(41.927407329, -87.70729439)		
3	JG496991	########	0000X W C	560	ASSAULT	SIMPLE	STREET	N	N	1832	42	08A	1176106	1905725	41.89667	-87.6286	(41.896671699, -87.628635323)		
4	JG497145	########	019XX W 4	051A	ASSAULT	AGGRAVAT	SIDEWALK	N	N	931	15	04A	1164331	1873509	41.80853	-87.6728	(41.808525157, -87.672792896)		
5	JG496701	########	025XX W E	502P	OTHER OF	FALSE / ST	STREET	N	N	2011	40	26	1158314	1935772	41.97951	-87.6932	(41.979505088, -87.693158103)		
6	JG484195	10/28/202	067XX S PA	810	THEFT	OVER $50(	APARTMEI	N	N	722	6	6	1173732	1860233	41.77189	-87.6387	(41.771890947, -87.638705659)		
7	JG483131	10/28/202	057XX N KI	1320	CRIMINAL	TO VEHICL	STREET	N	N	1711	39	14	1152676	1937956	41.98561	-87.7138	(41.985611859, -87.713834343)		
8	JG498494	########	089XX S CF	560	ASSAULT	SIMPLE	SIDEWALK	N	N	413	7	08A	1193055	1846244	41.73305	-87.5683	(41.733053891, -87.568330657)		
9	JG496575	########	037XX N S(	860	THEFT	RETAIL TH	SMALL RET	Y	N	1922	44	6	1166304	1924930	41.94959	-87.6641	(41.949586612, -87.664085689)		
10	JG427641	09/17/202	001XX W 1	820	THEFT	$500 AND	STREET	N	N	512	9	6	1177160	1835662	41.70439	-87.6269	(41.704388397, -87.626879123)		
11	JG365961	########	002XX W N	530	ASSAULT	AGGRAVAT	SMALL RET	N	N	122	34	04A	1174636	1900346	41.88194	-87.6342	(41.881944424, -87.634195294)		
12	JG496115	########	076XX S M	820	THEFT	$500 AND	STREET	N	N	621	17	6	1170966	1854231	41.75548	-87.649	(41.755481563, -87.649019949)		
13	JG496955	########	049XX N M	320	ROBBERY	STRONG A	SIDEWALK	N	N	1623	45	3	1139338	1932332	41.97043	-87.763	(41.970433391, -87.763029002)		
14	JG541270	12/14/202	072XX S C(	486	BATTERY	DOMESTIC	APARTMEI	Y	Y	324	7	08B	1189632	1857146	41.76305	-87.5805	(41.763052784, -87.580521082)		
15	JG501047	########	008XX E HY	620	BURGLARY	UNLAWFU	APARTMEI	N	N	223	20	5	1182731	1871378	41.80227	-87.6054	(41.802269632, -87.605372566)		
16	JG496779	########	013XX W 9	051A	ASSAULT	AGGRAVAT	SCHOOL -	N	N	2213	21	04A	1169194	1841762	41.7213	-87.6559	(41.721303358, -87.655873595)		
17	JG496296	########	0000X E RA	890	THEFT	FROM BUI	SPORTS AF	N	N	111	34	6	1176904	1901295	41.8845	-87.6258	(41.884497529, -87.625838595)		

FIG: 3.3 Representation of Dataset

Dataset Attributes:

- CASE#: This is likely a unique identifier for each reported crime case.
- DATE OF OCCURRENCE: This indicates the date and time when the crime occurred.
- BLOCK: It specifies the block where the crime occurred, giving a general location.
- IUCR: The Illinois Uniform Crime Reporting code, which is a standardized code representing the type of crime reported.
- PRIMARY DESCRIPTION: This provides a general category describing the primary type of crime committed.
- SECONDARY DESCRIPTION: Further details about the crime, providing additional information beyond the primary description.
- LOCATION DESCRIPTION: Describes the location where the crime occurred, providing contextual information.
- ARREST: Indicates whether an arrest was made in connection with the reported crime.
- DOMESTIC: Specifies whether the crime is categorized as domestic.
- BEAT: Refers to the police beat where the crime occurred, indicating a specific area for patrolling.
- WARD: Indicates the ward in which the crime occurred, providing a political division for city representation.
- FBI CD: FBI Crime Classification Code, providing additional categorization for the type of crime.
- X COORDINATE: The X coordinate of the location where the crime occurred, typically used in spatial analysis.
- Y COORDINATE: The Y coordinate of the location where the crime occurred,

used in conjunction with X coordinate for spatial analysis.

- LATITUDE: The latitude of the location where the crime occurred.
- LONGITUDE: The longitude of the location where the crime occurred.
- LOCATION: This provides the latitude and longitude coordinates in parentheses, which represent the exact geographic location of the crime.

3.2.4 WORKFLOW

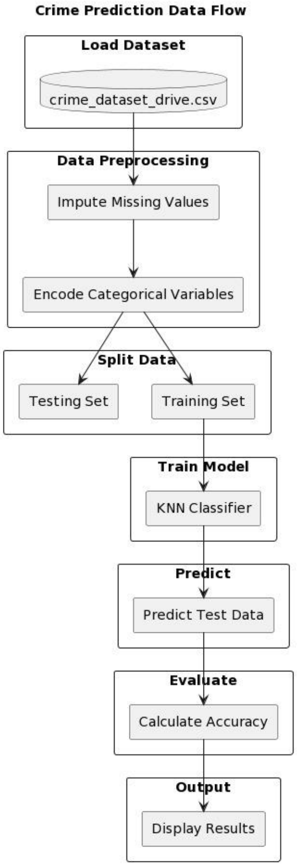


FIG: 3.4 Functional Requirements working

3.2.5 SYSTEM ARCHITECTURE

The system architecture for our project involves collecting crime data from data.govern.in spanning from January 2023 to January 2024. This dataset serves as the foundation for training our machine learning model using the K Nearest Neighbors (KNN) algorithm. Our goal is to enhance prediction accuracy to over 85% by refining the KNN approach. By reducing code complexity, we aim to streamline the model training process and provide law enforcement agencies with insights into potential crime patterns. The system will offer predictions on the likelihood of the next two crime types based on the historical data analysis.

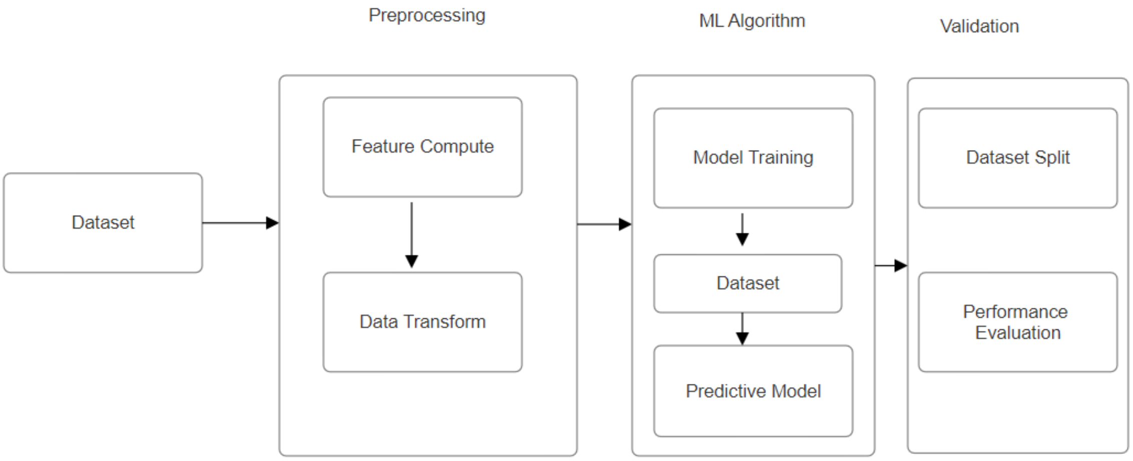


FIG: 3.5 Representation of System Architecture

3.2.6 MODULE

In software development, a module refers to a self-contained unit of code that performs a specific set of functions or tasks. Modules are designed to be reusable, maintainable, and easily understandable components of a software system. They help in organizing code into logical units, making it easier to manage and scale larger projects.

Modules can vary in size and complexity, ranging from small utility functions to larger components that encapsulate entire subsystems. They can be implemented in various programming languages and can interact with each other through well-defined interfaces

.

3.2.6.1 User Modules:

1. **Data Collection Module**: This module is responsible for fetching data from the data source, in your case, data.govern.in. It might involve web scraping or API calls to retrieve the required dataset.
2. **Data Preprocessing Module**: Before feeding the data into the machine learning model, it needs to be preprocessed. This includes tasks like handling missing values, encoding categorical variables, scaling numerical features, and splitting the dataset into training and testing sets.
3. **Feature Engineering Module**: This module focuses on creating new features from the existing ones or selecting the most relevant features to improve the performance of the model. Techniques like feature scaling, dimensionality reduction, or creating interaction terms can be applied here.
4. **Model Training Module**: Here, you'll implement the K Nearest Neighbors (KNN) algorithm to train the machine learning model. This involves feeding the preprocessed data into the algorithm, tuning hyperparameters if necessary, and training the model.
5. **Model Evaluation Module**: After training the model, it's important to evaluate its performance. This module calculates various metrics like accuracy, precision, recall, and F1 score to assess how well the model is performing. Cross-validation techniques can also be applied here to ensure robustness.
6. **Prediction Module**: Once the model is trained and evaluated, this module is responsible for making predictions on new data. It takes input data, preprocesses it, and passes it through the trained model to predict the likelihood of crime occurrence and the type of crime.

3.2.6.2 System Modules:

1. **Data Storage Module**: This module handles the storage of collected data, preprocessed data, and trained models. It might involve using databases or file systems to store and retrieve data efficiently.
2. **Logging and Monitoring Module**: For tracking the progress of the project and diagnosing issues, a logging and monitoring module is essential. It logs various events, errors, and performance metrics during data collection, preprocessing, model training, and evaluation stages.
3. **User Interface Module**: This module provides an interface for users to interact with the system. It could be a web-based dashboard, a desktop application, or a

command-line interface where users can input data, view predictions, and access other functionalities of the system.

4. **Deployment Module**: Once the model is trained and evaluated, it needs to be deployed into a production environment where it can be used to make real-time predictions. This module handles the deployment process, including setting up servers, deploying the model, and integrating it with other systems if necessary.

3.3 IDENTIFICATION OF NEED

In contemporary society, the rise in criminal cases poses significant challenges to law enforcement agencies, leading to a backlog of pending cases. Effectively controlling the increase in crimes is crucial to maintaining public safety and ensuring the efficient functioning of law enforcement agencies. One approach to address this challenge is to leverage machine learning models to predict crime patterns. By analyzing historical crime data, these models can identify potential patterns or trends in criminal behavior, aiding law enforcement in proactive intervention and crime prevention.

While several machine learning algorithms exist for crime prediction, such as Naïve Bayes and Decision Trees, there is a need to explore more accurate and efficient methods. K Nearest Neighbors (KNN) algorithm presents a promising approach due to its ability to classify data based on similarity measures. By training a model using historical crime data and employing the KNN algorithm, it is possible to enhance prediction accuracy and identify crime patterns more effectively.

Previous efforts utilizing the KNN algorithm have achieved a prediction rate of 75%. However, there is a clear opportunity to improve upon this performance and achieve a prediction rate of over 85%. Additionally, by focusing on the KNN algorithm, the aim is to reduce code complexity and streamline the implementation process.

To facilitate this project, a dataset spanning from January 2003 to January 2004 has been collected from data.govern.in. This dataset will serve as the foundation for training and testing the machine learning model. The project will aim to not only enhance prediction accuracy but also provide insights into the likelihood of future crime types based on historical data analysis. Ultimately, the goal is to empower law enforcement agencies with advanced predictive capabilities to combat crime more effectively.

3.4 FEASIBILITY STUDY

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Economical Feasibility
- Operational Feasibility
- Technical Feasibility

3.4.1 ECONOMIC FEASIBILITY

Economic feasibility of predicting the next two crimes using machine learning lies in its potential to optimize resource allocation and enhance efficiency in law enforcement efforts. By accurately forecasting the types of crimes expected in a particular area, authorities can strategically deploy personnel and resources, thereby minimizing costs associated with reactionary measures and maximizing the impact of preventive interventions. Moreover, by proactively addressing crime patterns, the societal and economic costs of criminal activities, such as property damage, loss of productivity, and psychological distress, can be mitigated. Thus, investing in machine learning for crime prediction not only holds promise for improving public safety but also yields long-term economic benefits by fostering a safer and more secure environment conducive to economic prosperity.

3.4.2 OPERATIONAL FEASIBILITY

Operational feasibility in predicting the next two crimes using machine learning lies in its practical implementation within the existing law enforcement infrastructure. By harnessing machine learning algorithms on crime data sets, law enforcement agencies can proactively identify patterns and trends in criminal activity. This predictive capability enables them to allocate resources more efficiently, deploy preventive measures, and enhance surveillance in high-risk areas. Additionally, it empowers agencies to prioritize investigations and interventions based on the likelihood of specific crimes occurring, thereby optimizing their response strategies. Ultimately, operational feasibility rests on the integration of machine learning tools into the daily operations of law enforcement, and targeted crime prevention efforts.

3.4.2 TECHNICAL FEASIBILITY

The technical feasibility of predicting the next two crimes using machine learning lies in the availability of comprehensive crime datasets and the development of robust predictive models. By leveraging advanced algorithms such as decision trees, neural networks, or support vector machines, along with appropriate feature engineering techniques, it's possible to identify meaningful patterns in crime data. Utilizing historical crime data, spatial and temporal features, socio-economic indicators, and other relevant variables, machine learning models can be trained to forecast the likelihood of specific types of crimes occurring in a given area within a defined timeframe. However, the accuracy and reliability of such predictions depend on the quality and quantity of data, as well as the efficacy of the chosen algorithms and model evaluation methods. Nevertheless, with the advancements in machine learning and data analytics, coupled with the availability of sophisticated tools and technologies, the technical feasibility of predicting future crimes in a meaningful manner is increasingly achievable.

3.5 PROJECT PLANNING

1. **Project Overview**

Briefly describe the project, which involves analyzing crime data and predicting the next two crimes using machine learning algorithms.

2**. Objectives**

- Define the main objectives of the project:
- Analyze historical crime data to identify patterns and trends.
- Develop machine learning models to predict the occurrence of future crimes.
- Provide actionable insights for law enforcement agencies to enhance crime prevention strategies.

3. **Data Collection**

- Identify sources for obtaining historical crime data, such as public databases or government agencies.
- Gather relevant datasets containing information about past crimes, including location, type, date, and time.

4. **Data Preprocessing**

- Clean the collected data to handle missing values, outliers, and inconsistencies.
- Perform feature engineering to extract meaningful features from the raw data.
- Explore and visualize the data to gain insights into crime patterns and distributions.

5. **Model Selection**

- Evaluate different machine learning algorithms suitable for crime prediction tasks, such as decision trees, random forests, or neural networks.
- Choose appropriate evaluation metrics, such as accuracy, precision, recall, or F1-score, to assess model performance.

6. **Model Development**

- Split the preprocessed data into training and testing sets for model training and evaluation.
- Train machine learning models on the training data using chosen algorithms.
- Fine-tune model hyperparameters and optimize performance using cross-validation techniques.

7. **Model Evaluation**
    - Evaluate the trained models on the testing data using selected evaluation metrics.
    - Compare the performance of different models to identify the most accurate and reliable predictors of future crimes.
8. **Deployment**
    - Deploy the trained machine learning models into a production environment for real-time prediction.
    - Develop a user interface or application for law enforcement personnel to input relevant data and receive crime predictions.
9. **Testing and Validation**
    - Conduct thorough testing and validation of the deployed models to ensure reliability and accuracy in real-world scenarios.
    - Validate predictions against actual crime occurrences to measure the effectiveness of the models.
10. **Documentation and Reporting**
    - Document the entire project, including data sources, preprocessing steps, model development, evaluation results, and deployment procedures.
    - Prepare a comprehensive report summarizing the findings, insights, and recommendations for stakeholders and interested parties.

3.6 SOFTWARE SPECIFIC REQUIREMENTS

For your project, you'll need to focus on developing a robust machine learning model using the K Nearest Neighbors (KNN) algorithm to predict crime patterns. This model aims to achieve an accuracy rate of over 85%, surpassing previous attempts. To accomplish this, you'll utilize a dataset collected from data.gov.in spanning from January 2023 to January 2024. The software requirements for this project include programming languages and libraries for machine learning such as Python with libraries like scikit-learn for implementing the KNN algorithm efficiently. Additionally, you'll need tools for data preprocessing, visualization, and evaluation to ensure the model's effectiveness and accuracy. The goal is not only to predict crime patterns accurately but also to streamline the code complexity for better maintenance and scalability. By leveraging advanced techniques and optimizing the KNN algorithm, you aim to contribute significantly to crime prediction and prevention efforts.

3.6.1FUNCTIONAL REQUIREMENTS

To address the rising criminal cases and alleviate the backlog of pending cases, we propose the development of a Machine Learning Model utilizing the K Nearest Neighbors (KNN) Algorithm. Our aim is to surpass the existing prediction rate of 75% achieved by previous models and achieve an accuracy of over 85%. By leveraging this approach, we intend to identify potential crime patterns to aid law enforcement agencies in proactive crime prevention. The functional requirements for this project include:

1. Dataset Collection: Gather comprehensive data from data.gov.in spanning from January 2023 to January 2024 to ensure a robust training set.
2. Preprocessing: Cleanse and preprocess the dataset to remove noise and inconsistencies, ensuring high-quality data for model training.
3. Feature Selection: Identify relevant features such as location, time, demographics, and crime type to train the model effectively.
4. Model Training: Implement the KNN Algorithm to train the machine learning model using the collected dataset.
5. Evaluation Metrics: Utilize appropriate evaluation metrics such as accuracy, precision, recall, and F1-score to assess the model's performance.

6. Hyperparameter Tuning: Optimize the KNN model by fine-tuning hyperparameters to enhance predictive accuracy.
7. Predictive Analysis: Develop a user-friendly interface to showcase the predicted likelihood of the occurrence of the next two crime types based on historical data.
8. Code Optimization: Streamline the codebase to reduce complexity and enhance efficiency, ensuring smooth integration and maintenance of the system.

3.6.2NON-FUNCTIONAL REQUIREMENTS

For this project, focusing on improving prediction accuracy and reducing code complexity is crucial. We aim to enhance the KNN algorithm's performance to achieve an accuracy rate exceeding 85% using the dataset obtained from data.gov.in spanning from January 2023 to January 2024. Here are some non-functional requirements to consider:

1. Scalability: Ensure the model can handle large volumes of data efficiently as the dataset grows over time.
2. Performance: Optimize the algorithm to deliver fast and accurate predictions, allowing law enforcement agencies to act swiftly.
3. Reliability: The model should consistently provide accurate predictions under various circumstances and datasets.
4. Security: Implement measures to safeguard sensitive crime data and ensure privacy compliance.
5. Ease of Use: Simplify the deployment and usage of the model for law enforcement personnel, possibly through a user-friendly interface.
6. Flexibility: Design the system to accommodate future updates or changes in crime data collection methods or algorithms.

3.7 DATA VISUALIZATION

Crime visualization totally allowed to show the dataset analysis in a visualized format like in way that a normal user can easily go through it. In detail way to plot the graphs or make them to understand by bar graphs etc… The analysis can be done through

- In a period of time the number of crimes are might committed.
- By taking a city to observe the number of crimes over all crime types
- Ratio of taking them in custody that is the ratio of arrest in police records.
- By considering the different locations observing the committed crimes.
- Details of crimes that are happen majorly in city.

-

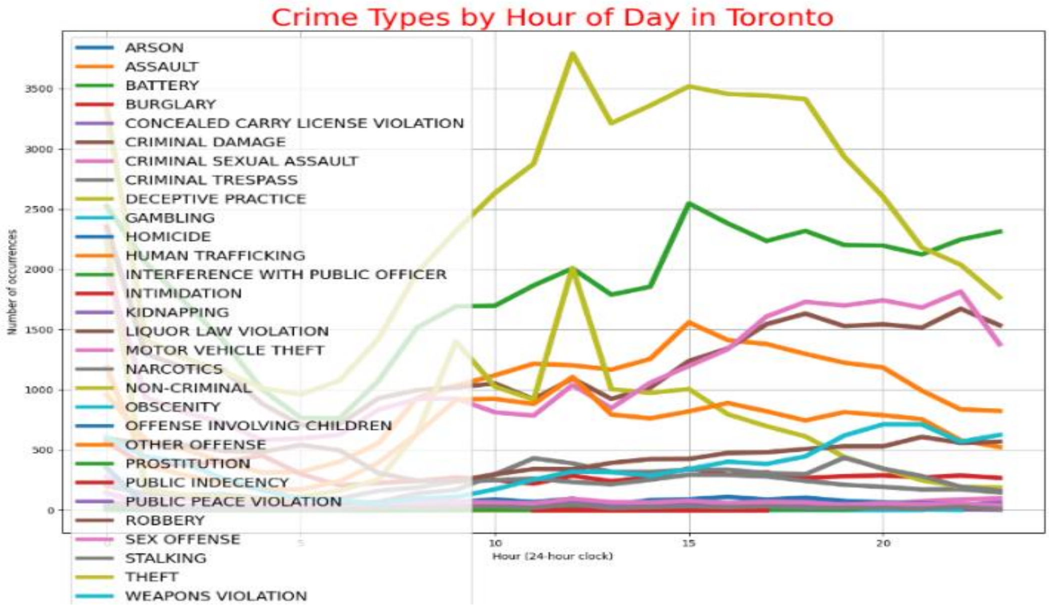
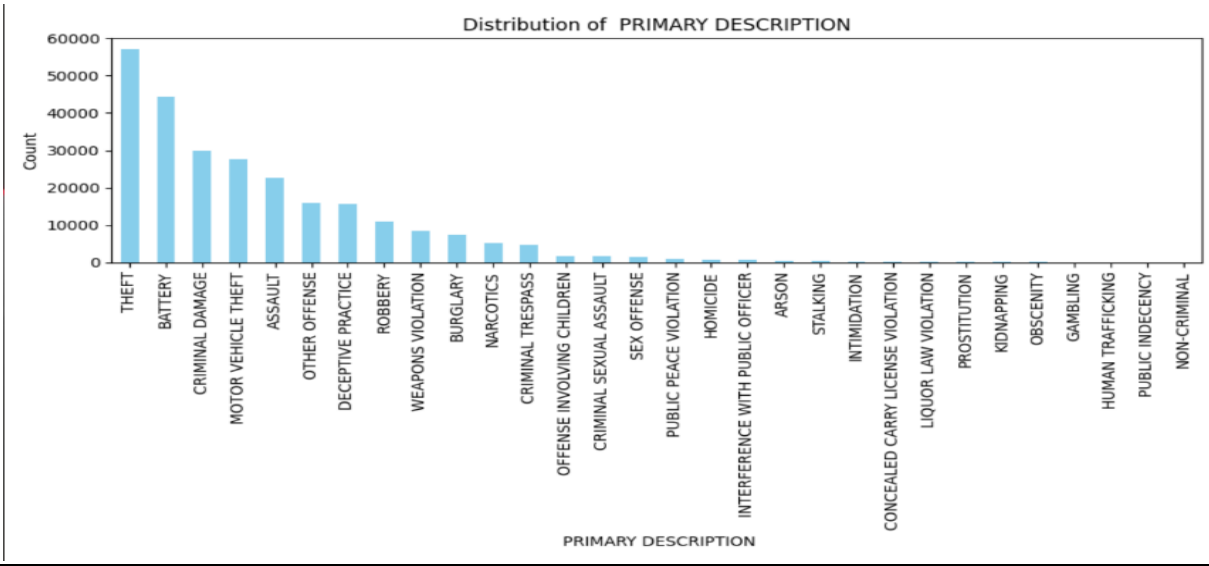


Fig.3.6 Major Crime Indicator

By analyzing the previous crimes we representing in the form of a bar graph which crime indicates mostly. Here Theft is the major crime indicator, secondary is



the Battery Indicator etc… Here we represented x-axis as crime names and y-axis as crimes count. The bar graph represents the major crimes to minor crimes from left to right.

Fig.3.7 Crime Types by Hours

Here we have used line graph , In one graph we combined all lines into one graph. Here each line represents one type of crime, each line represents the crime and also it represents the time on that day of occurrence which is in 24-hr format.

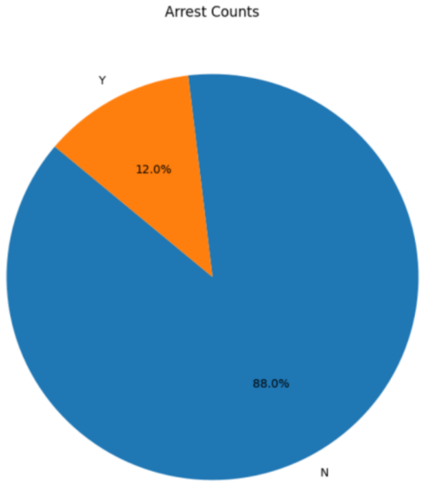


Fig.3.8 Pie Chart representing the Arrest counts

By using Pie chart we have represented the arrest counts.If we see in past 2023 Jan-2024 Jan only 12% cases accused has been arrest remaining 88% cases are still in pending.

Chapter 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS

- Minimum: Intel i3 processor, 4GB RAM, 250GB hard disk
- Operating Systems: Windows or Linux

4.2SOFTWARE REQUIREMENTS

- Operating System : Windows 10/11
- Coding Language and Version : Python (3.11)
- Tool : Visual Studio
- Documentation : Ms-Office

Chapter 5

SYSTEM DESIGN

5.1 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

5.1.1 USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

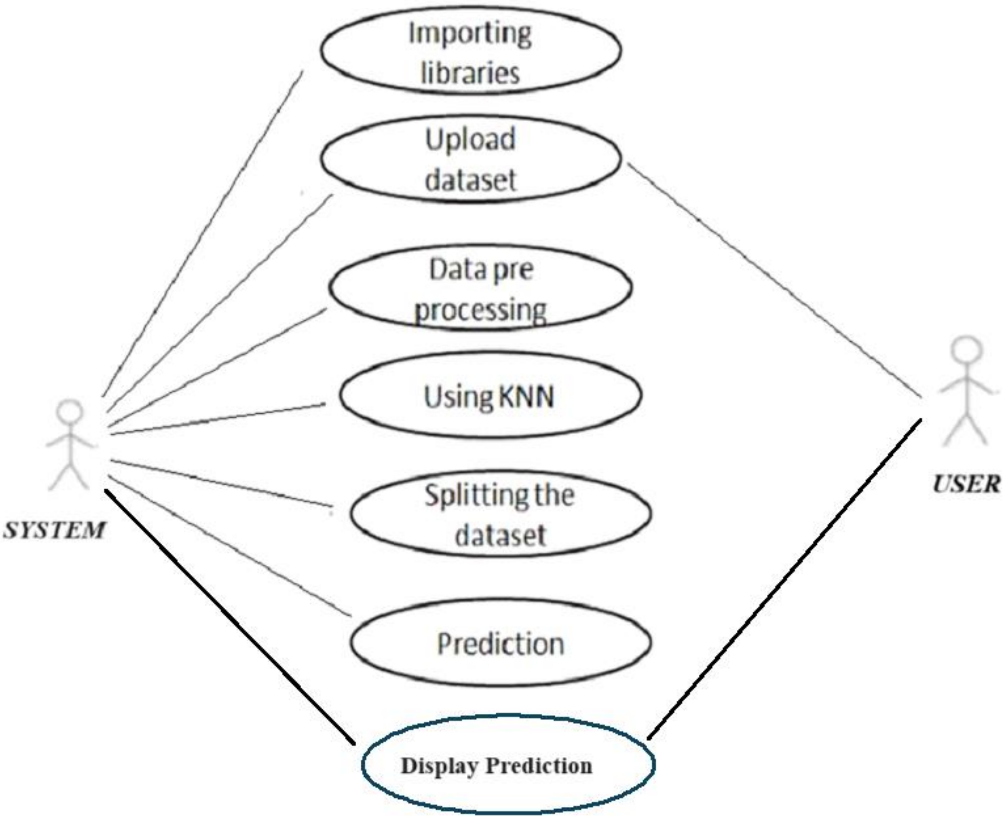


FIG:5.1Representation of Use Case Diagram

5.1.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

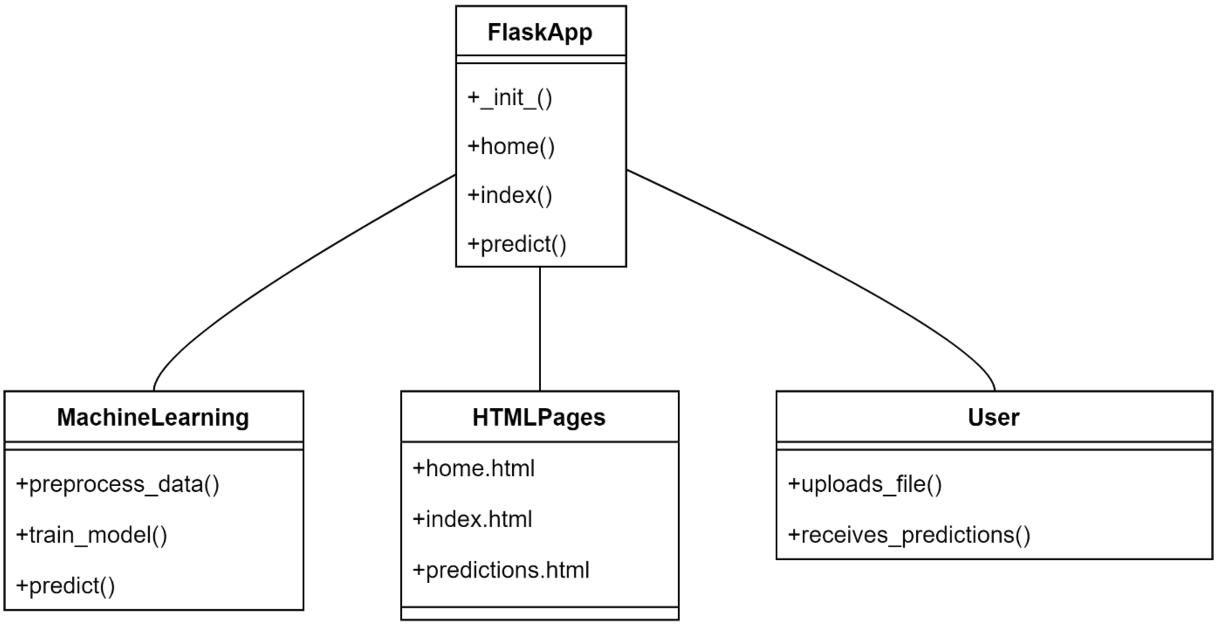


FIG:5.2 Representation of Class Diagram

5.1.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

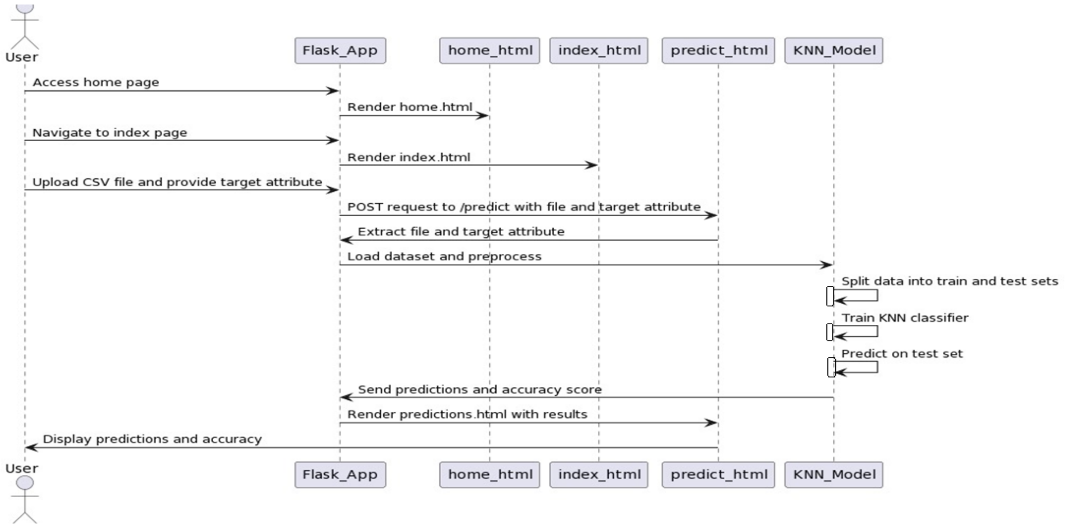

User
Flask_App
home_html
index_html
predict_html
KNN_Model
Access home page
Render home.html
Navigate to index page
Render index.html
Upload CSV file and provide target attribute
POST request to /predict with file and target attribute
Extract file and target attribute
Load dataset and preprocess
Split data into train and test sets
Train KNN classifier
Predict on test set
Send predictions and accuracy score
Render predictions.html with results
Display predictions and accuracy
User
Flask_App
home_html
index_html
predict_html
KNN_Model

FIG:5.3 Representation of Sequence Diagram

5.1.4 DFD DIAGRAM

DFD stands for Data Flow Diagram. It is a visual representation of the flow of data within a system, illustrating how data is input, processed, stored, and outputted. DFD diagrams consist of various components including processes, data stores, data flows, and external entities.

- Processes: Represent functions or operations performed on the data, such as calculations, transformations, or validations.
- Data Stores: Represent where data is stored within the system, such as databases, files, or repositories.
- Data Flows: Represent the movement of data between processes, data stores, and external entities.
- External Entities: Represent external sources or destinations of data, such as users, systems, or other organizations interacting with the system.

DFD diagrams are commonly used in systems analysis and design to model and understand the flow of data in a system, aiding in the identification of system requirements, communication between stakeholders, and the design of system architectures.

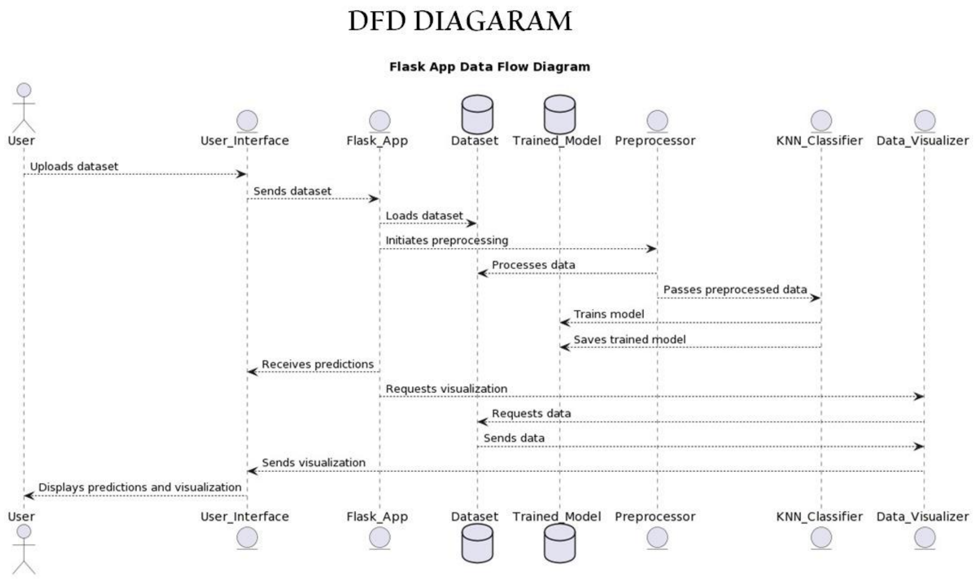


FIG:5.4 Representation of DFD Diagram

5.1.5 ACTIVITY DIAGRAM

An activity diagram is a graphical representation of the flow of activities within a system, process, or workflow. It visually depicts the sequence of actions, decisions, and interactions among components or actors involved in completing a specific task or achieving a particular goal. Activity diagrams are part of the Unified Modeling Language (UML) and are commonly used in software engineering and business process modeling to illustrate the dynamic aspects of a system. They help stakeholders understand the order of execution, decision points, and concurrency within a process, facilitating communication and analysis of complex systems.

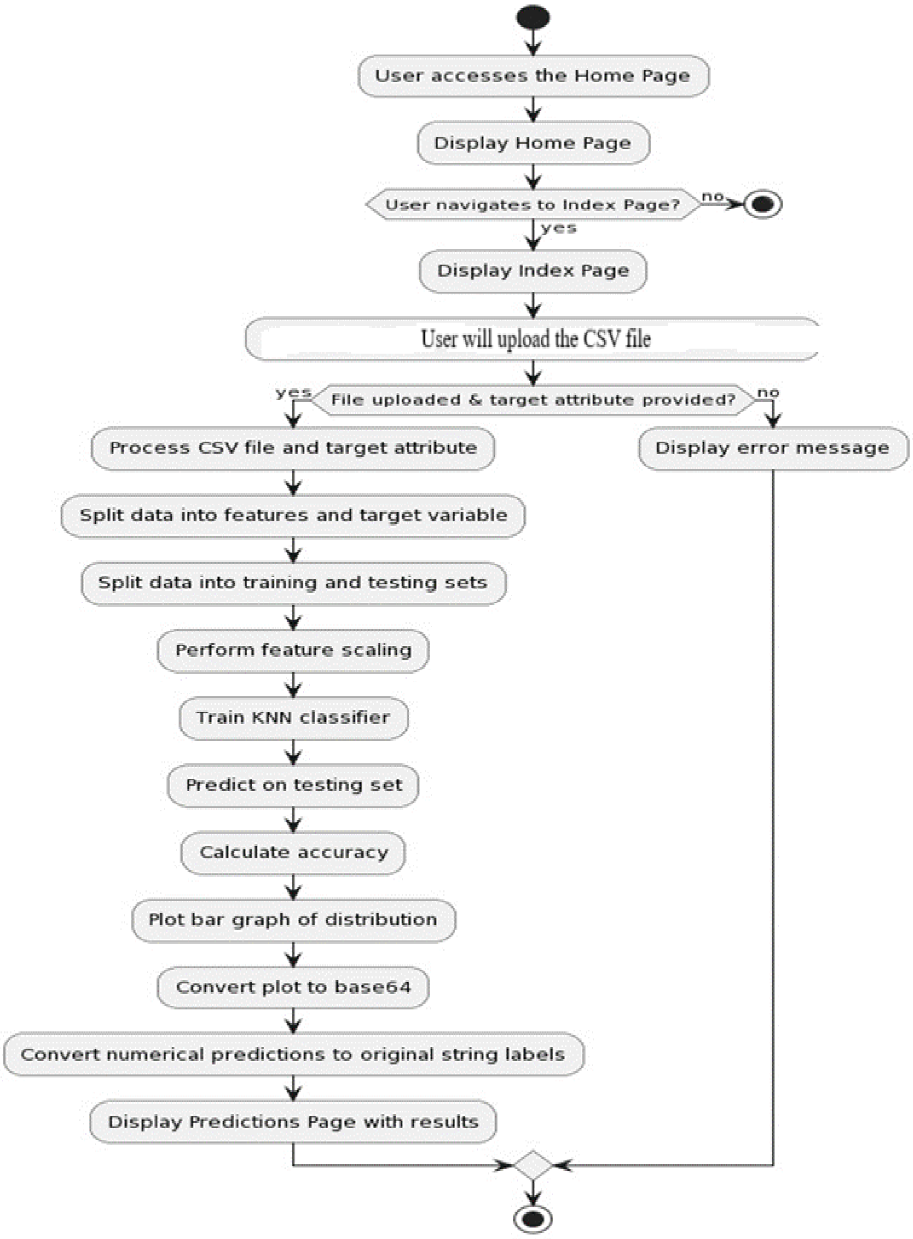

User accesses the Home Page
Display Home Page
User navigates to Index Page?
no
yes
Display Index Page
User will upload the CSV file
yes
File uploaded & target attribute provided?
no
Process CSV file and target attribute
Display error message
Split data into features and target variable
Split data into training and testing sets
Perform feature scaling
Train KNN classifier
Predict on testing set
Calculate accuracy
Plot bar graph of distribution
Convert plot to base64
Convert numerical predictions to original string labels
Display Predictions Page with results

FIG:5.5 Representation of Activity Diagram

5.2 INPUT AND OUTPUT DESIGN

Input Design

The input design is the link between the information system and the user. It comprises the developing specifications and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining privacy. Input Design considered the following things:

- ➢ What data should be given as input?
- ➢ How the data should be arranged or coded?
- ➢ The dialog guides the operating personnel in providing input.
- ➢ Methods for preparing input validations and steps to follow when error occur.

Objective

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output.

It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well-thought-out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively.
2. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements
3. Select methods for presenting information.
4. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

Chapter 6

SYSTEM IMPLEMENTATION

Since computers do not communicate like humans, coding acts as a translator. Code converts human input into numerical sequence that computers understand. Once computers receive these messages, they complete assigned tasks such as changing font colours or centring an image. "Programming languages provide the rules for building websites, app and other computers – based technologies."

Coding tells a machines which actions to performs and how to complete tasks. Programming languages provide the rules for building website, apps, and others computer-based technologies. Each programming language helps accurately communicates with machines.

6.1 PROJECT PREREQUESITIES

Front-end Technologies

- HTML: Display content on web pages
- CSS: Creates beautiful styles

Back-end Technologies

- PYTHON: Programming Language

HTML: HTML, or Hypertext Markup Language, is the standard markup language for creating web pages and web applications. It defines the structure and content of a webpage by using a system of tags and attributes. HTML documents are comprised of elements, which are enclosed in opening and closing tags and may contain content or other nested elements. These elements are organized hierarchically, forming a tree-like structure known as the Document Object Model (DOM). HTML provides a variety of tags for structuring text, creating links, embedding multimedia, defining forms, and more. It is a fundamental technology used in conjunction with Cascading Style Sheets (CSS) and JavaScript to build and design interactive and visually appealing websites.

FEATURES:

- ✓ User Friendly
- ✓ Media Support
- ✓ Platform Independent
- ✓ Free and open-source

CSS: Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in HTML or XML. It enables web developers to control the layout, appearance, and formatting of web pages across different devices and screen sizes. CSS allows for the separation of content from presentation, making it easier to maintain and update websites. By applying CSS rules to HTML elements, developers can specify properties such as color, font, spacing, and positioning, thus achieving a consistent and visually appealing design. CSS also supports responsive design techniques, enabling web pages to adapt to various viewport sizes and orientations, improving usability and accessibility.

FEATURES:

- ✓ Time saving
- ✓ Easy maintenance
- ✓ Multiple frameworks
- ✓ Offline support

PYTHON: Python is a high-level, interpreted programming language known for its simplicity and readability, making it ideal for beginners and experienced developers alike. It offers a vast ecosystem of libraries and frameworks for various purposes, including web development, data analysis, machine learning, and automation. Python's syntax emphasizes code readability and clarity, promoting productivity and collaboration among programmers. Its versatility and cross-platform compatibility make it a popular choice for a wide range of applications, from scripting small tasks to building complex software systems. Overall, Python's ease of use, extensive community support, and broad applicability make it a powerful tool for solving diverse programming challenges.

FEATURES:

- ✓ Ridiculously fast
- ✓ Reassuringly secure
- ✓ Exceedingly scalable
- ✓ Portable
- ✓ SEO friendly
- ✓ Rapid development

Why Python

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated procedurally, an object-orientated way or a functional way.

Machine Learning

Explicitly programmed, aids in making predictions or decisions with the assistance of sample historical data, or training data. For the purpose of developing predictive models, machine learning brings together statistics and computer science. Algorithms that learn from historical data are either constructed or utilized in machine learning. The performance will rise in proportion to the quantity of information we provide.

How does Machine Learning work

A machine learning system builds prediction models, learns from previous data, and predicts the output of new data whenever it receives it. The amount of data helps to build a better model that accurately predicts the output, which in turn affects the accuracy of the predicted output.

Let's say we have a complex problem in which we need to make predictions. Instead of writing code, we just need to feed the data to generic algorithms, which build the logic based on the data and predict the output. Our perspective on the issue has changed as a result of machine learning. The Machine Learning algorithm's operation is depicted in the following block diagram:

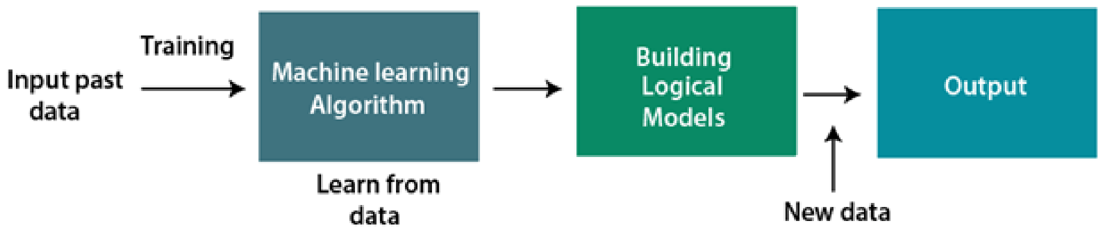


FIG: 6.1 How does Machine Learning work

Features of Machine Learning:

- ✓ Machine learning uses data to detect various patterns in a given dataset.
- ✓ It can learn from past data and improve automatically.
- ✓ It is a data-driven technology.
- ✓ Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Types of Machine Learning

- ➢ Supervised Machine Learning
- ➢ Unsupervised Machine Learning
- ➢ Reinforcement Machine Learning

1. **Supervised Machine Learning:**

Supervised learning is a type of machine learning in which the algorithm is trained on the labeled dataset. It learns to map input features to targets based on labeled training data. In supervised learning, the algorithm is provided with input features and corresponding output labels, and it learns to generalize from this data to make predictions on new, unseen data.

There are two main types of supervised learning:

- Regression: Regression is a type of supervised learning where the algorithm learns to predict continuous values based on input features. The output labels in regression are continuous values, such as stock prices, and housing prices. The different regression algorithms in machine learning are: Linear Regression, Polynomial Regression, Ridge Regression, Decision Tree Regression, Random Forest Regression, Support Vector Regression, etc
- Classification: Classification is a type of supervised learning where the algorithm learns to assign input data to a specific category or class based on input features. The output labels in classification are discrete values. Classification algorithms can be binary, where the output is one of two possible classes, or multiclass, where the output can be one of several classes. The different Classification algorithms in machine learning are: Logistic Regression, Naive Bayes, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), etc

2. Unsupervised Machine Learning:

Unsupervised learning is a type of machine learning where the algorithm learns to recognize patterns in data without being explicitly trained using labeled examples. The goal of unsupervised learning is to discover the underlying structure or distribution in the data.

There are two main types of unsupervised learning:

- Clustering: Clustering algorithms group similar data points together based on their characteristics. The goal is to identify groups, or clusters, of data points that are similar to each other, while being distinct from other groups. Some popular clustering algorithms include K-means, Hierarchical clustering, and DBSCAN.
- Dimensionality reduction: Dimensionality reduction algorithms reduce the number of input variables in a dataset while preserving as much of the original information as possible. This is useful for reducing the complexity of a dataset and making it easier to visualize and analyze. Some popular dimensionality reduction algorithms include Principal Component Analysis (PCA), t-SNE, and Autoencoders.

3. Reinforcement Machine Learning

Reinforcement learning is a type of machine learning where an agent learns to interact with an environment by performing actions and receiving rewards or penalties based on its actions. The goal of reinforcement learning is to learn a policy, which is a mapping from states to actions, that maximizes the expected cumulative reward over time.

There are two main types of reinforcement learning:

- Model-based reinforcement learning: In model-based reinforcement learning, the agent learns a model of the environment, including the transition probabilities between states and the rewards associated with each state-action pair. The agent then uses this model to plan its actions in order to maximize its expected reward. Some popular model-based reinforcement learning algorithms include Value Iteration and Policy Iteration.
- Model-free reinforcement learning: In model-free reinforcement learning, the agent learns a policy directly from experience without explicitly building a model of the environment. Some popular model-free reinforcement learning algorithms.

6.2 SAMPLE CODE

Source:

app.py

```
from flask import Flask from flask import Flask, render_template,
request, redirect, url_for
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
import io
import base64
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from datetime import datetime
import warnings
from warnings import filterwarnings
filterwarnings('ignore')
app = Flask(__name__, static_url_path='/static/styles.css')

@app.route('/')
def home():
    return render_template('home.html')
@app.route('/index')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    # Get uploaded file and target attribute name
    file = request.files['file']
    target_attribute = request.form['target_attribute']
    target_attribute1=target_attribute



    # Load the dataset
    df = pd.read_csv(file)
    df1=df[target_attribute]
    unique_labels = df[target_attribute].unique()
    def convert_to_standard_format(date_str):
        formats = ['%m-%d-%Y %H:%M', '%m/%d/%Y %I:%M:%S %p', '%m/%d/%Y
%I:%M:%S %p', '%m-%d-%Y %H:%M:%S']
        for fmt in formats:
            try:
                dt = datetime.strptime(date_str, fmt)
                return dt.strftime('%Y-%m-%d %H:%M:%S')
            except ValueError:
                pass
        return None
    # Convert the 'DATE OF OCCURRENCE' column to standard format and
add as a new column
    df['Standard Dates'] = df['DATE  OF
OCCURRENCE'].apply(convert_to_standard_format)
    # Convert 'DATE OF OCCURRENCE' to datetime format
    df['Standard Dates'] = pd.to_datetime(df['Standard Dates'])

    # Extract hour from 'DATE OF OCCURRENCE'
    df['hour'] = df['Standard Dates'].dt.hour
    # Group data by hour and crime type
    hour_crime_group = df.groupby(['hour', ' PRIMARY
DESCRIPTION']).size().reset_index(name='Total')
    # Plotting
    fig, ax = plt.subplots(figsize=(15, 10))
    for crime_type, data in hour_crime_group.groupby(' PRIMARY
DESCRIPTION'):
        ax.plot(data['hour'], data['Total'], label=crime_type,
linewidth=5)
    # Customize plot
    ax.set_xlabel('Hour (24-hour clock)')
    ax.set_ylabel('Number of occurrences')
    ax.set_title('Crime Types by Hour of Day in Toronto', color='red',
fontsize=25)
    ax.grid(linestyle='-')
    leg = plt.legend(fontsize='x-large')
    leg_lines = leg.get_lines()
    plt.setp(leg_lines, linewidth=4)
    # Convert plot to base64
    img = io.BytesIO()
    plt.savefig(img, format='png')
    img.seek(0)
    plot_url1 = base64.b64encode(img.getvalue()).decode()

    # Plot pie chart for arrest counts
    arrest_counts = df['ARREST'].value_counts()

    # Plot pie chart
    plt.figure(figsize=(8, 8))
    plt.pie(arrest_counts, labels=arrest_counts.index,
autopct='%1.1f%%', startangle=140)
    plt.title('Arrest Counts')
    img_arrest_counts = io.BytesIO()
    plt.savefig(img_arrest_counts, format='png')
    img_arrest_counts.seek(0)
    plot_url_arrest_counts =
base64.b64encode(img_arrest_counts.getvalue()).decode()


    # Preprocessing
    non_numeric_columns = df.select_dtypes(exclude=['number']).columns
    label_encoder = LabelEncoder()
    for column in non_numeric_columns:
        df[column] = label_encoder.fit_transform(df[column])
    df.fillna(df.mean(), inplace=True)
    x = df.drop([target_attribute,'Standard Dates','WARD','BEAT'],
axis=1)

    y = [target_attribute]

    # Split the data into training and testing sets
    x_train, x_test, y_train, y_test = train_test_split(x, df[y],
test_size=0.3, random_state=57)

    # Feature scaling
    scaler = StandardScaler()
    x_train_scaled = scaler.fit_transform(x_train)
    x_test_scaled = scaler.transform(x_test)
    # Initialize KNN classifier
    knn_classifier = KNeighborsClassifier()

    # Train the classifier
    knn_classifier.fit(x_train_scaled, y_train)


    # Predict on the testing set
    y_pred = knn_classifier.predict(x_test_scaled)
    # Calculate accuracy
    accuracy = accuracy_score(y_test, y_pred)

    # Plot bar graph
    plt.figure(figsize=(10, 6))
    df1.value_counts().plot(kind='bar', color='skyblue')
    plt.xlabel(target_attribute1)
    plt.ylabel('Count')
    plt.title('Distribution of ' + target_attribute1)
    plt.tight_layout()

    # Convert plot to base64
    img = io.BytesIO()
    plt.savefig(img, format='png')
    img.seek(0)
    plot_url = base64.b64encode(img.getvalue()).decode()

    # Create a dictionary to map numerical predictions to original
string labels
    #unique_labels = df[y[0]].unique()
    prediction_mapping = {i: label for i, label in
enumerate(unique_labels)}

    # Convert numerical predictions to original string labels using the
dictionary
    y_pred_original = [prediction_mapping[prediction] for prediction in
y_pred]
    first_two_predictions = y_pred_original[:2]
    return render_template('predictions.html',
predictions=first_two_predictions, accuracy=accuracy,
plot_url=plot_url,plot_url1=plot_url1,
                           plot_url_bar_graph=plot_url_arrest_counts)


if __name__ == '__main__':
    app.run(debug=True)
```

Home.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    html,* {
      box-sizing: border-box;
      margin: 0;
      padding: 0;
    }

    body {
      background-image:
url('https://free4kwallpapers.com/uploads/originals/2020/07/06/black-
red-texture-wallpaper.jpg');
      background-size: cover;
      background-position: center;
      background-repeat: no-repeat;
      color: white; /* Change text color to white */
    }

    .container {
      display: flex;
      flex-wrap: wrap;
      justify-content: center;
      gap: 16px;
      padding: 16px;
    }
    .par{
        font-family: Georgia, 'Times New Roman', Times, serif;
        font-style: Times New Roman;
    }

    .card {
      width: calc(25% - 16px);
      background-color: rgba(255, 255, 255, 0.5); /* Smokey white
background */
      border-radius: 8px;
      overflow: hidden;
    }

    .card .image {
      width: 100%;
      height: 200px;
      object-fit: cover;
    }

    .card .content {
      padding: 16px;
      text-align: center;
    }

    .card .content h2 {
      margin-bottom: 8px;
    }

    .box {
      display: block;
      background-color: antiquewhite;
    }

    .carousel-container {
      width: 80%; /* Adjust width as needed */
      height: 100%;
      overflow-x: auto; /* Enable horizontal scrolling */
      overflow-y: hidden; /* Hide vertical scrolling */
      margin: auto; /* Center the carousel */
      margin-top: 20px; /* Add space from the top */
    }
    .carousel {
      display: flex;
      align-items: center;
      justify-content: flex-start; /* Adjust alignment for flex-start
*/
      font-family: Arial;
    }

    .carousel__list {
      display: flex;
      list-style: none;
      padding: 0;
      margin: 0;
      perspective: 300px;
    }

    .carousel__item {
      display: flex;
      flex-shrink: 0; /* Prevent items from shrinking */
      align-items: center;
      justify-content: center;
      color: black;
      font-size: 10px;
      width: 250px;
      height: 250px;
      margin-right: 10px; /* Adjust spacing between items */
      border-radius: 12px;
      box-shadow: 0px 2px 8px 0px rgba(50, 50, 50, 0.5);
      position: relative;
      transition: all .3s ease-in;
    }

    /* Adjust positioning for the first item */
    .carousel__item:first-child {
      margin-left: 10px; /* Adjust spacing for the first item */
    }

    /* Adjust positioning for the last item */
    .carousel__item:last-child {
      margin-right: 0; /* Remove margin for the last item */
    }

    /* Add more styles for the rest of the images */

    .carousel__text {
      position: absolute;
      bottom: 10px;
      left: 10px;
      color: white;
      font-size: 16px;
      font-weight: bold;
      opacity: 0;
      transition: opacity 0.3s ease;
    }

    .carousel__item:hover .carousel__text {
      opacity: 1;
    }

    /* Set dimensions for images */
    .carousel__item img {
      width: 100%;
      height: 100%;
      object-fit: cover;
      border-radius: 12px;
    }
    .button-container {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 10vh; /* This ensures the button is vertically centered
*/
    }

    /* Style for the button */
    .button {
        background-color: #4CAF50;
        border: none;
        color: white;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 24px;
        padding: 20px;
        cursor: pointer;
        border-radius: 10px;
        transition: background-color 0.3s;
    }

    /* Styling for the hover effect */
    .button:hover {
        background-color: #45a049;
    }
  </style>
</head>
<body>
  <br/>
  <br/>
  <center>
    <h1 style="font-size: 65px;">Crime Prediction</h1>
    <br/>
    <div class="box">
      <p style="font-size: 30px; font-style: italic;color:
black;">"Unlock insights from crime data using advanced machine
learning techniques."</p>
    </div>
  </center>
  <br/>
  <div class="carousel-container">
    <div class="carousel">
      <ul class="carousel__list">
        <!-- Adjust images and content for each carousel item -->
        <li class="carousel__item">
          <img src="https://media.istockphoto.com/photos/crime-scene-
picture-id979502094?k=6&m=979502094&s=170667a&w=0&h=VO3IsPrUj89xUI-
sSfWlVlgcsdbZfDt3D-oln_8nJYU=" alt="Slide 1">
```

```
          <span class="carousel__text">
            <h2>Murder</h2>
            <p>Crime data analysis to uncover patterns and insights,
shedding light on the dynamics and factors influencing murder
occurrences.</p>
          </span>
        </li>
        <li class="carousel__item">
          <img src="https://www.factinate.com/wp-
content/uploads/2017/07/robbery_05767900.jpg" alt="Slide 2">
          <span class="carousel__text">
            <h2>Robbery</h2>
            <p>Explore predictive insights on robbery patterns through
advanced machine learning analysis of crime data.</p>
          </span>
        </li>
        <li class="carousel__item">
          <img
src="https://cdn.cfr.org/sites/default/files/styles/open_graph_article/
public/image/2021/02/human-trafficking-protest.jpg" alt="Slide 3">
          <span class="carousel__text">
            <h2>Human Trafficking</h2>
            <p>Explore insights on human trafficking patterns and
trends through machine learning analysis.</p>
          </span>
        </li>
        <li class="carousel__item">
          <img
src="https://tse2.mm.bing.net/th?id=OIP.ny2hRfCUHAlgK4w1HJgmoQHaEK&pid=
Api&P=0&h=220" alt="Slide 4">
          <span class="carousel__text">
            <h2>Assault</h2>
            <p>Predictive machine learning models applied to assault
crime data for analysis and pattern recognition.</p>
          </span>
        </li>
        <li class="carousel__item">
          <img src="https://www.shouselaw.com/wp-
content/uploads/2020/11/battery.jpg" alt="Slide 5">
```

```
     <span class="carousel__text">
      <h2>Battery</h2>
      <p>Using machine learning, analyze crime data to predict
instances of battery.</p>
     </span>
    </li>
    <li class="carousel__item">
     <img src="https://www.wklaw.com/wp-
content/uploads/2013/11/Cyberstalking-PC-646.9.jpg" alt="Slide 6">
     <span class="carousel__text">
      <h2>Stalking</h2>
      <p>Utilizing machine learning for crime data analysis to
predict and prevent instances of stalking.</p>
     </span>
    </li>
    <li class="carousel__item">
     <img
src="https://www.mlive.com/resizer/raHMpCSH5y9XooKA0etZk33Mplk=/1280x0/
smart/advancelocal-adapter-image-
uploads.s3.amazonaws.com/image.mlive.com/home/mlive-
media/width2048/img/grpress/news_impact/photo/aljpg-
8147e1978ec3cfb6.jpg" alt="Slide 7">
     <span class="carousel__text">
      <h2>Weapons Violation</h2>
      <p>Applying machine learning techniques to analyze patterns
and predict instances of weapons violation in crime data.</p>
     </span>
    </li>
    <li class="carousel__item">
     <img src="https://www.shouselaw.com/wp-
content/uploads/2022/02/child-kidnapping.jpg" alt="Slide 8">
     <span class="carousel__text">
      <h2>Kidnapping</h2>
      <p>Applying machine learning techniques to analyze patterns
and predict instances of kidnapping in crime data.</p>
     </span>
    </li>
   </ul>
  </div>
  </div>
  <br/>

   <center>
    <div class="par">
    <p style="font-size: 30px; font-style: TimesNewRoman;">If you are
interested in predicting the next crime, please upload a .csv file and
provide the target attribute you want to predict for the next
crime.</p>

    <br/>

    <p style="font-size: 30px; font-style: italic;">If you want to
upload your own dataset then Proceed here</p>
</div>
</center>
    <br/>
    <br/>

  </center>
  <div class="button-container">
    <!-- Button with onclick event to navigate to the index page -->
    <button class="button" onclick="window.location.href =
'index';">GO</button>
</div>

  <script>
    const carouselList = document.querySelector('.carousel__list');
    const carouselItems = document.querySelectorAll('.carousel__item');
    const elems = Array.from(carouselItems);

    carouselList.addEventListener('click', function (event) {
      var newActive = event.target;
      var isItem = newActive.closest('.carousel__item');

      if (!isItem ||
newActive.classList.contains('carousel__item_active')) {
        return;
      };
      update(newActive);
    });

    const update = function(newActive) {
      const newActiveIndex =
Array.from(newActive.parentNode.children).indexOf(newActive);

      elems.forEach((item, index) => {
        const itemPos = index - newActiveIndex;
        item.dataset.pos = itemPos;
      });
    };
  </script>
</body>
</html>
```

Index.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='styles.css') }}">
    <title>Index Page</title>
    <style>
        .box{
            display: block;
            background-color: aquamarine;
        }
        .card1 {
            border: 2px solid #ccc;
            border-radius: 10px;
            padding: 20px;
            width: 300px; /* Adjust width as needed */
            margin: 0 auto;
            background-color: rgba(255, 255, 255, 0.5); /* Transparent
background */
        }

        .form-group1 {
            margin-bottom: 20px;
        }

        label {
            display: block;
            margin-bottom: 5px;
        }

        input[type="file"],
        input[type="text"],
        input[type="submit"] {
            width: 100%;
            padding: 10px;
            border-radius: 5px;
            border: 1px solid #ccc;
            box-sizing: border-box;
        }

        input[type="submit"] {
            background-color: #007bff;
            color: white;
            cursor: pointer;
        }

        input[type="submit"]:hover {
            background-color: #0056b3;
        }

    body {
        background-image:
url('https://free4kwallpapers.com/uploads/originals/2020/07/06/black-
red-texture-wallpaper.jpg');
        background-size: cover; /* Cover the entire viewport */
        background-position: center;
        background-repeat: no-repeat;
        margin: 0; /* Remove default body margin */
        padding: 0; /* Remove default body padding */
        height: 100vh; /* Set body height to viewport height */

    }
</style>

    </style>

</head>
<body>
    <center><h1 style="font-size: 30px; font-style: italic; color:
whitesmoke;"></h1></center>

<div class="card1">
    <form action="{{ url_for('predict') }}" method="post"
enctype="multipart/form-data">
        <div class="form-group1">
            <label for="file">Upload CSV File:</label>
            <input type="file" name="file" id="file" accept=".csv">
        </div>
        <div class="form-group1">
            <label for="target_attribute">Target Attribute:</label>
            <input type="text" name="target_attribute"
id="target_attribute" placeholder="Enter target attribute">
        </div>

        <input type="submit" value="Predict">
    </form>
</div>

</body>
</html>
```

Prediction.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='styles.css') }}">
    <title>Predictions</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            background-image:
url('https://free4kwallpapers.com/uploads/originals/2020/07/06/black-
red-texture-wallpaper.jpg');
            background-size: cover; /* Cover the entire viewport */
            background-position: center;
            background-repeat: no-repeat;
            margin: 0; /* Remove default body margin */
            padding: 0; /* Remove default body padding */
            height: 100vh;
            text-decoration-color: beige;
        }
        h1 {
            color: white;
        }
        .prediction {
            margin-bottom: 10px;
            font-size: 18px;
            color: white; /* Change prediction text color to white */
        }
        .accuracy {
            margin-bottom: 20px;
            font-size: 18px;
            color: white; /* Change accuracy text color to white */
        }
        .graph {
            margin-top: 20px;
        }
    </style>
</head>
```

```
<body>
    <h1>Predictions</h1>
    <div class="accuracy">Accuracy: {{ accuracy }}</div>
    <div class="prediction">Prediction 1: {{ predictions[0] }}</div>
    <div class="prediction">Prediction 2: {{ predictions[1] }}</div>
    <div class="graph">
        <img src="data:image/png;base64,{{ plot_url }}" alt="Bar
Graph">
    </div>
</body>
</html>
```

6.3 IMPLEMENTATION

Implementation is the process of translating a concept, plan, or design into tangible action or realization. It involves executing the intended strategy or solution through practical steps, such as coding software, assembling hardware, or deploying systems, to achieve the desired outcome. In various fields, including technology, business, and education, implementation is fundamental to turning ideas into practical solutions or products. It often entails careful planning, resource allocation, and execution to ensure that the final result aligns with the initial goals and requirements.

We have collected the dataset from officially at data.govern.in that we are using in our current project. This is information is maintained that is updated with every change by Chicago police department.

working on the project is followed by several steps they are

A. Collection of Information

We collected the dataset from Data.govern.in in .csv extension.Our dataset name is Crimes_-_One_year_prior_to_present dataset.

B. Data preprocessing

Our dataset consists of --- entries. First we have converted all attributes into numerical datatype using label encoder and then replaced all Null Values using median Strategy.

C. Feature selection

Generally feature selection plays a crustial role that is it used to be build the model. The attributes which are used for feature selection are Block, Location, case, X coordinate, Y coordinate, Latitude, Longitude.

D. Building and Training the model

Location and --- attributes are used for the training after feature selection and then the dataset is classified into x train, y train and x test, y test. S klearn is used to import the model algorithms[fit(x train, y train)].

E. Prediction

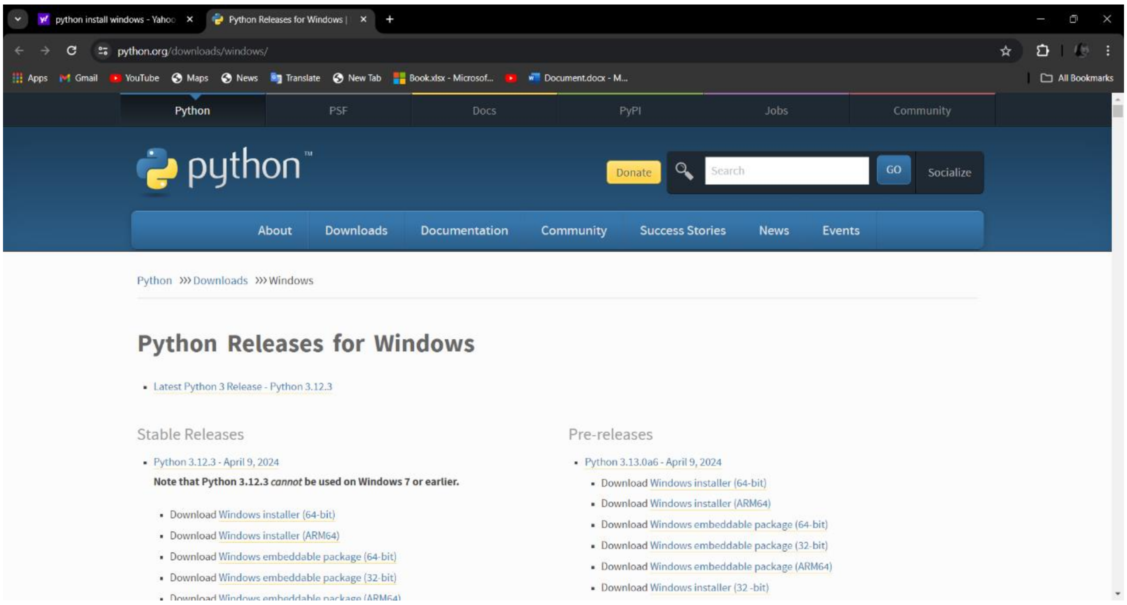
model.predict(xtest) is used to done the prediction after done the all previous process and build the model. accuracy_score is used to calculate the accuracy which is acutally imported from the metrics.accuracy_score(ytest, predicted). This is the process we generally used in prediction process.

F. Visualization

We represented the crime analysis in many ways like plotting graphs and represented in pie charts.

6.3.1PYTHON INSTALLATION AND VS CODE

We have to install the current latest version of python for the better and quick performance and best outcome



Python »» Downloads »» Windows

Python Releases for Windows

- Latest Python 3 Release - Python 3.12.3

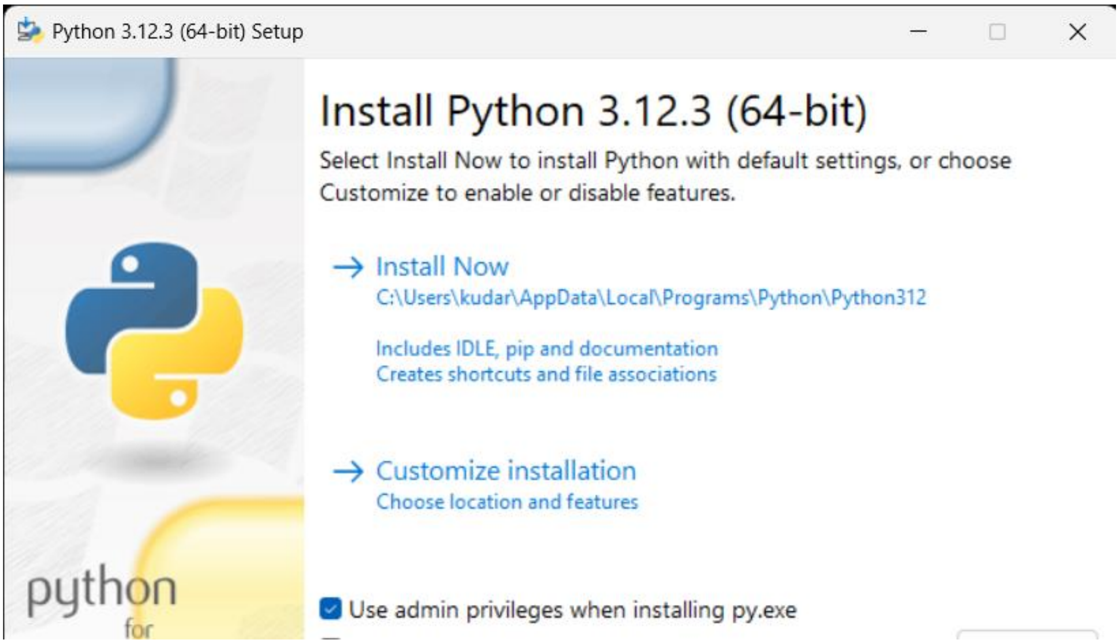
Stable Releases

- Python 3.12.3 - April 9, 2024
  Note that Python 3.12.3 *cannot* be used on Windows 7 or earlier.
  - Download Windows installer (64-bit)
  - Download Windows installer (ARM64)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (ARM64)

Pre-releases

- Python 3.13.0a6 - April 9, 2024
  - Download Windows installer (64-bit)
  - Download Windows installer (ARM64)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (ARM64)
  - Download Windows installer (32 -bit)

FIG: 6.2 Python download page



Python 3.12.3 (64-bit) Setup

Install Python 3.12.3 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

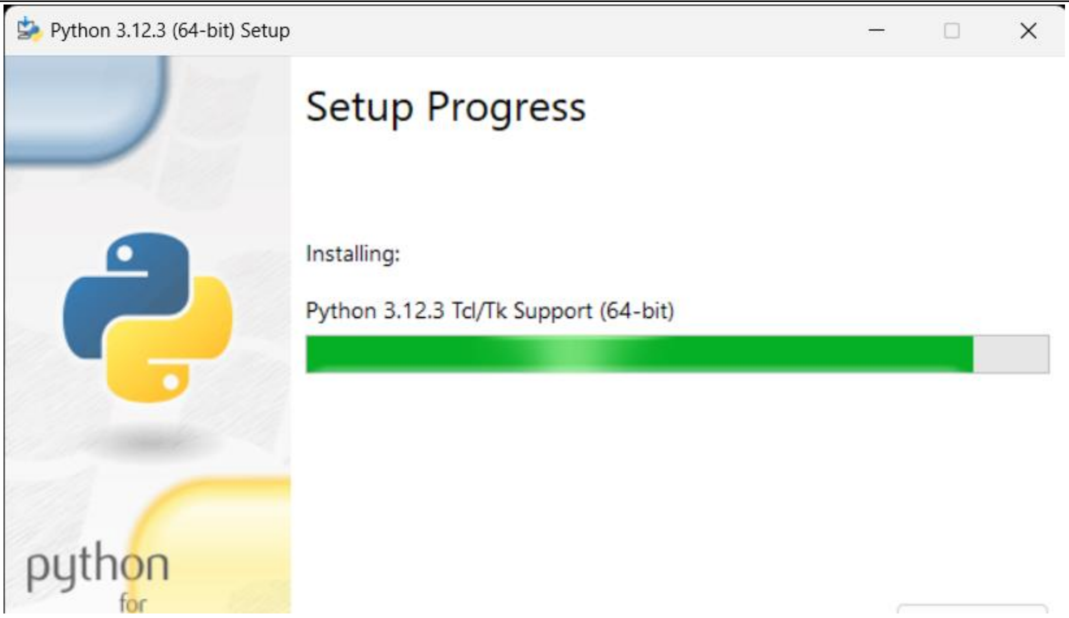
→ Install Now
C:\Users\kudar\AppData\Local\Programs\Python\Python312

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Use admin privileges when installing py.exe

FIG:6.3 Python install process

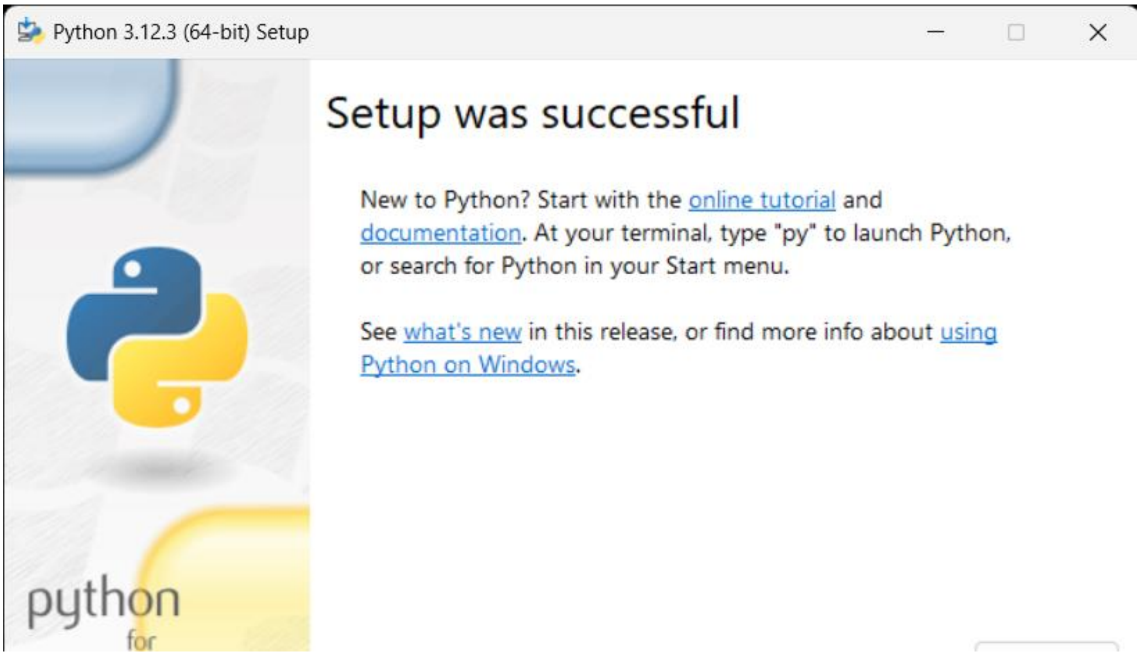


Setup Progress

Installing:

Python 3.12.3 Tcl/Tk Support (64-bit)

FIG: 6.4 Python setup progress in system



Setup was successful

New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

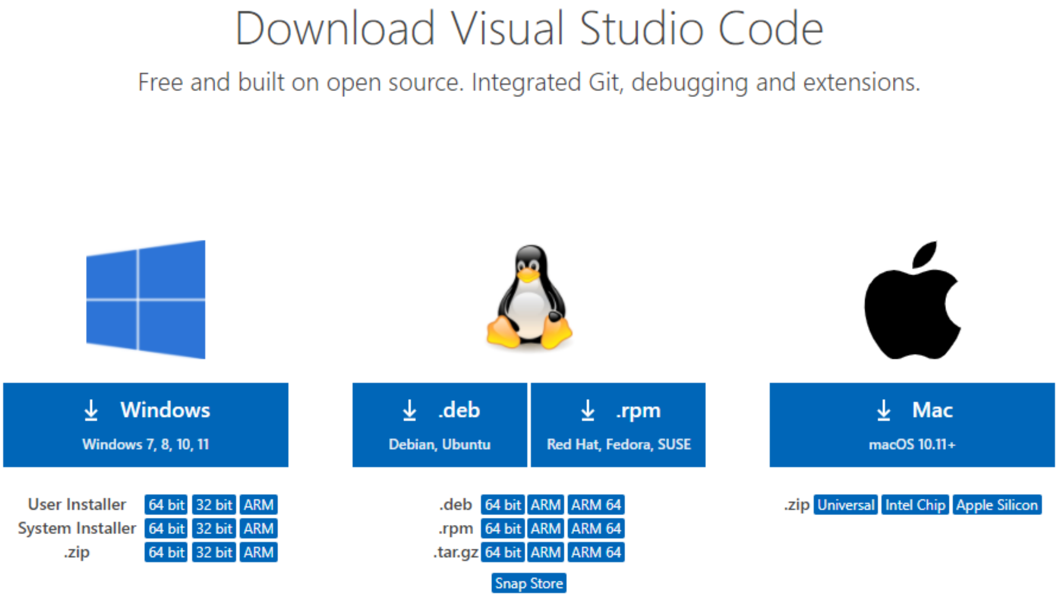
See what's new in this release, or find more info about using Python on Windows.

FIG:6.5 Setup was successful in system

Steps to Install Visual Studio Code on Windows

Step 1: Visit the [**Official Website**]() of the **Visual Studio Code** using any web browser like [Google Chrome](), [Microsoft Edge](), etc.

Step 2: Press the "**Download for Windows**" button on the website to start the download of the Visual Studio Code Application.



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

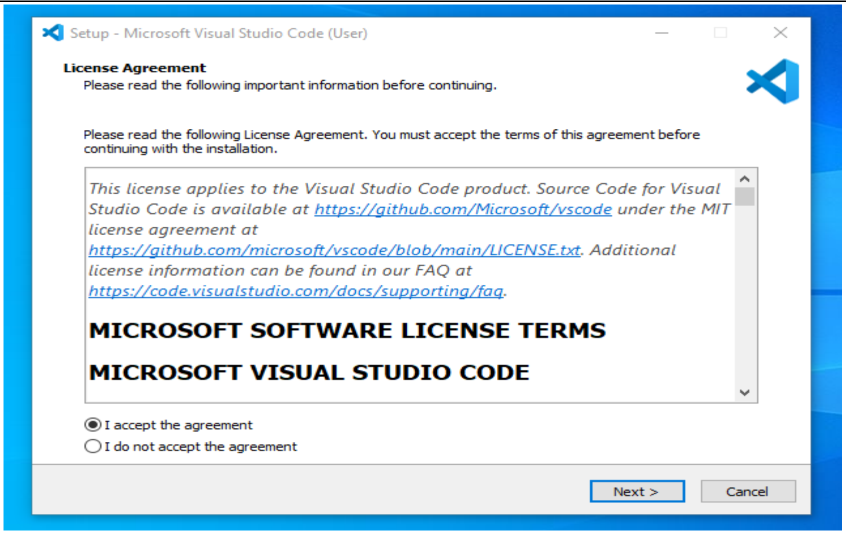
FIG:6.6 Visual Studio Code website

Step 3: When the download finishes, then the **Visual Studio Code Icon** appears in the downloads folder.

FIG:6.7 Visual Studio Code icon

Step 4: Click on the **Installer** icon to start the installation process of the Visual Studio Code.

Step 5: After the Installer opens, it will ask you to accept the terms and conditions of the Visual Studio Code. Click on **I accept the agreement** and then click the **Next** button.



Setup - Microsoft Visual Studio Code (User)

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

MICROSOFT SOFTWARE LICENSE TERMS

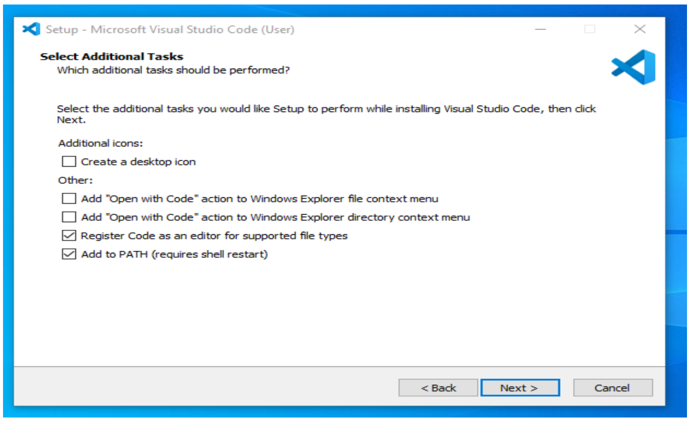
MICROSOFT VISUAL STUDIO CODE

● I accept the agreement

○ I do not accept the agreement

FIG:6.8 License aggrement

Step 6: Choose the location data for running the Visual Studio Code. It will then ask you to browse the location. Then click on the **Next** button.



Setup - Microsoft Visual Studio Code (User)

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

□ Create a desktop icon

Other:

□ Add "Open with Code" action to Windows Explorer file context menu

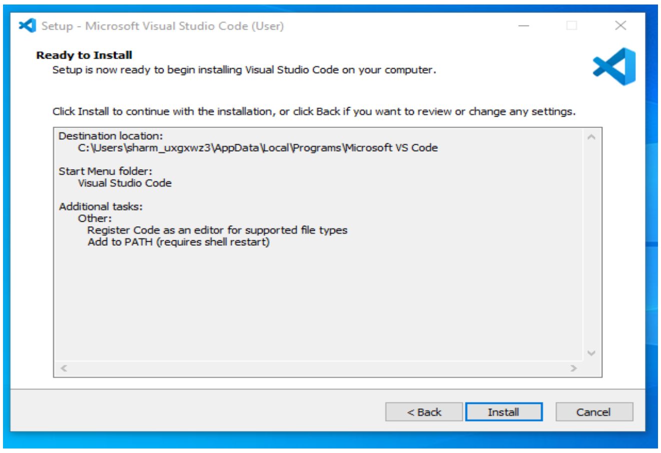
□ Add "Open with Code" action to Windows Explorer directory context menu

☑ Register Code as an editor for supported file types

☑ Add to PATH (requires shell restart)

FIG:6.9 The Location Data For Running The Visual Studio Code

Step 7: Then it will ask to begin the installation setup. Click on the **Install** button.



Setup - Microsoft Visual Studio Code (User)

Ready to Install

Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

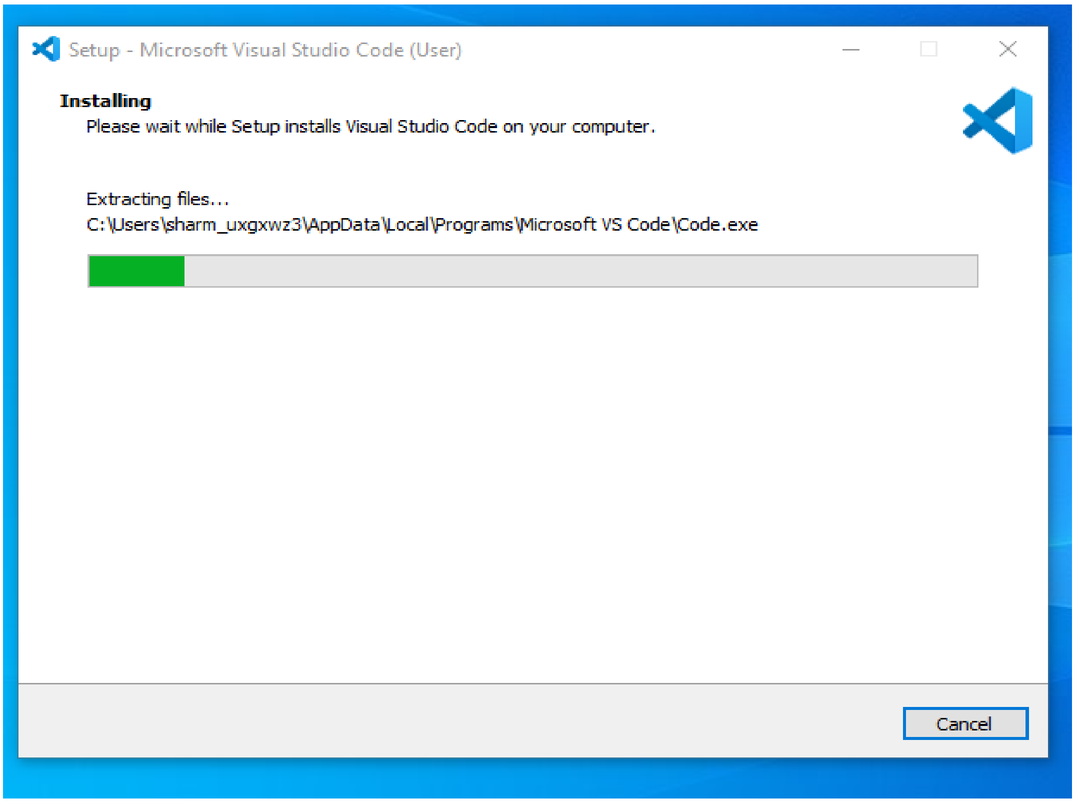
```
Destination location:
    C:\Users\sharm_uxgxwz3\AppData\Local\Programs\Microsoft VS Code

Start Menu folder:
    Visual Studio Code

Additional tasks:
    Other:
      Register Code as an editor for supported file types
      Add to PATH (requires shell restart)
```

FIG:6.10 Installation Setup of Visual Studio Code

Step 8: After clicking on Install, it will take about 1 minute to install the Visual Studio Code on your device.



Setup - Microsoft Visual Studio Code (User)

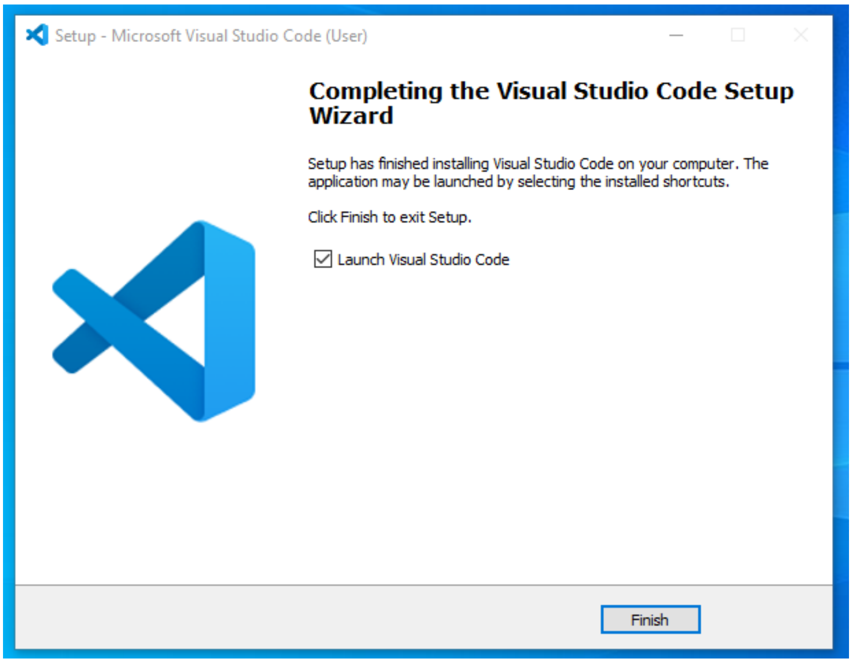
Installing

Please wait while Setup installs Visual Studio Code on your computer.

Extracting files...
C:\Users\sharm_uxgxwz3\AppData\Local\Programs\Microsoft VS Code\Code.exe

FIG:6.11 Installation of Visual Studio Code

Step 9: After the Installation setup for Visual Studio Code is finished, it will show a window like this below. Tick the “**Launch Visual Studio Code**” checkbox and then click **Next**.



Setup - Microsoft Visual Studio Code (User)

Completing the Visual Studio Code Setup Wizard

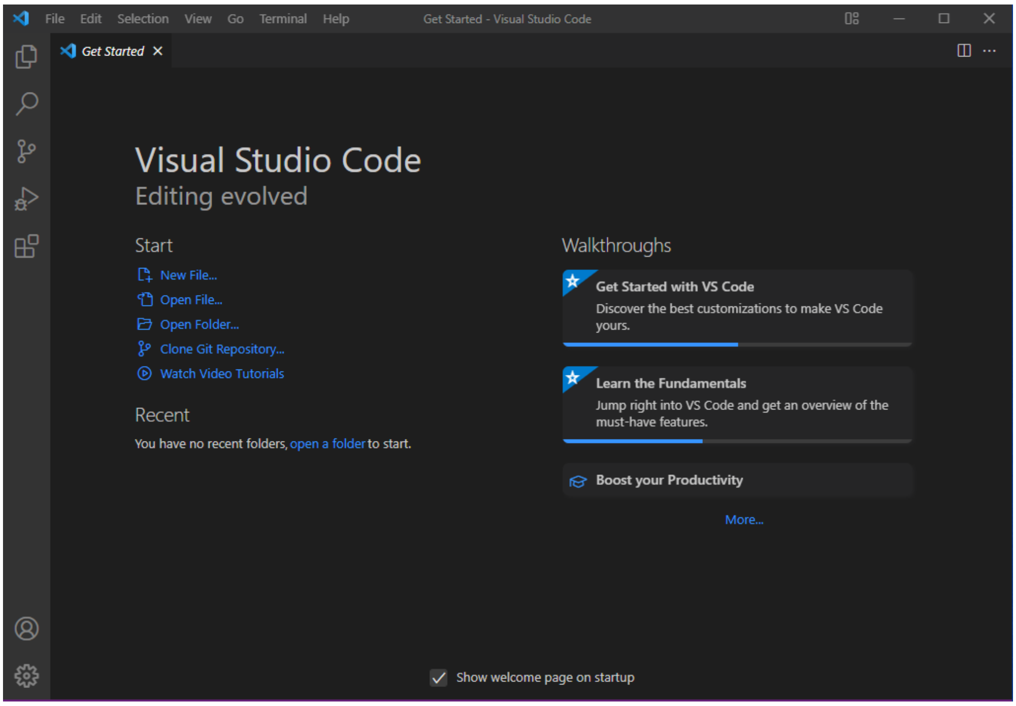
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch Visual Studio Code

FIG:6.12 Installation Completed

Step 10: After the previous step, the **Visual Studio Code window** opens successfully. Now you can create a new file in the Visual Studio Code window and choose a language of yours to begin your programming journey!



You have no recent folders, open a folder to start.

Discover the best customizations to make VS Code yours.

Jump right into VS Code and get an overview of the must-have features.

FIG:6.13 Visual Studio Code home page

So this is how we successfully installed **Visual Studio Code** on our Windows system. The steps mentioned in the above guideline can be used in any kind of Windows Browsers for **Downloading & Installation VS Code on Windows 11**. So, try these steps to get the **VS Code IDE** on your device as well.

63.1 STEPS TO RUN THE WEBSITE

1. **Homepage:**
   - Upon visiting the website, users land on the homepage.
   - The homepage contains a button or link labeled "Go" that redirects users to the index page when clicked.
2. **Index Page:**
   - After clicking "Go" on the homepage, users are directed to the index page.
   - On the index page, users see a form with options to upload a CSV file and specify a target variable.
   - Users select a CSV file from their local system using the file upload option.

- Users input the name of the target variable (e.g., the variable they want to predict) into a text field.
- After selecting the file and entering the target variable, users submit the form.

3. **Prediction Page:**

- Upon form submission, the website processes the uploaded CSV file and target variable.
- The prediction page is displayed, showing the accuracy of the predictions made by the machine learning model.
- Users see the two most recent predictions for crimes based on the uploaded data.
- Additionally, users see graphical visualizations such as:
- A line graph depicting the distribution of crime occurrences by hour of the day in Toronto.
- A pie chart showing the distribution of arrest counts.

• A bar graph illustrating the distribution of the target variable specified by the user

CHAPTER 7

TESTING

7.1TESTING

Testing is the systematic process of evaluating a product or system to verify whether it meets specified requirements and to identify any discrepancies between expected and actual outcomes. It involves executing the software or system under controlled conditions to detect defects, errors, or issues that may affect its functionality, reliability, performance, or security. Testing encompasses various activities, methodologies, and techniques, ranging from manual inspection to automated testing, all aimed at ensuring the quality and correctness of the product before it is released to users.

7.1.1 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 tests. Each test type addresses a specific testing requirement.

7.1.2TYPES OF TESETING

7.1.2.1 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.

7.1.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.1.2.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box

7.1.2.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under test is treated as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.1.2.7 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

7.1.2.8 TESTING RESULTS

All the test cases mentioned above passed successfully. No defects encountered.

7.2 TESTING METHODOLOGIES

The following are the Testing Methodologies:

- Unit Testing.
- Integration Testing.
- User Acceptance Testing.
- Output Testing.
- Validation Testing.

7.2.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually, and the module interfaces are verified for the consistency with design specification. All-important processing path are tested for the expected results. All error handling paths are also tested.

7.2.2 INTEGRATION TESTING

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1)Top-Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2) Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom-up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program.

The bottom-up approach tests each module individually and then each module is module is integrated with a main module and tested for functionality.

7.2.3 USER ACCEPTANCE TESTING

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

7.2.4 OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

7.2.5 VALIDATION TESTING

Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error message. The individual modules are checked for accuracy and what it must perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

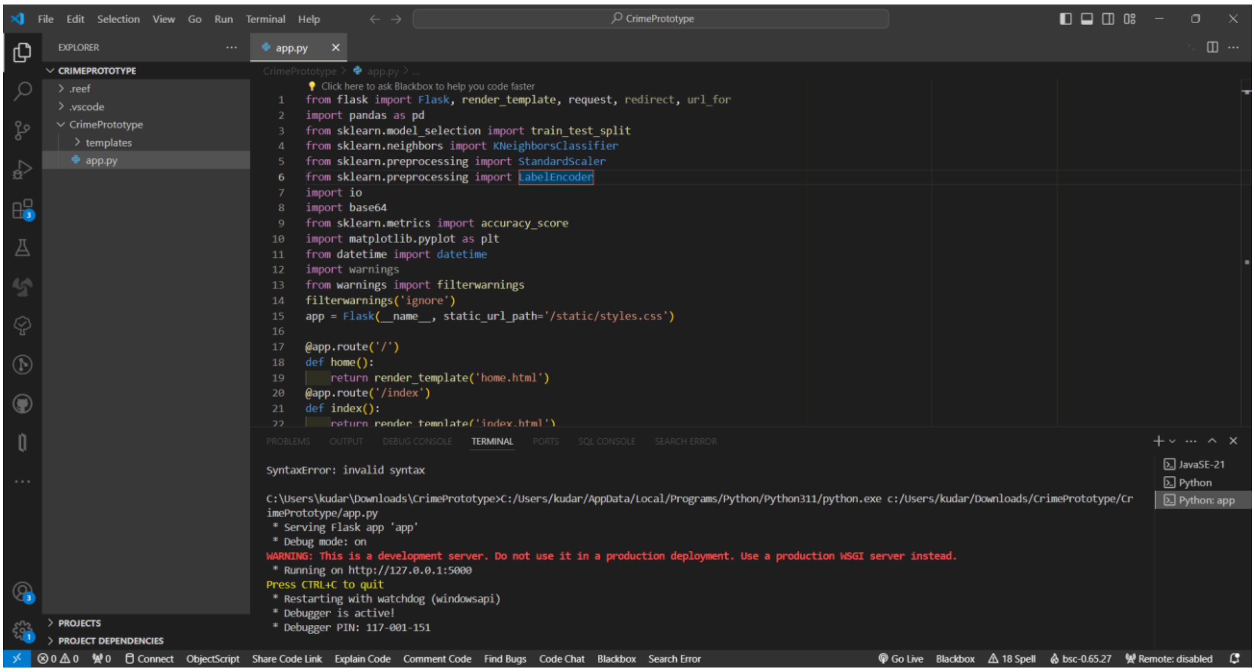
Live test data are those that are extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves. It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that would show how the system would perform for the typical processing requirement, if the live data entered is in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true system test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, makes possible the testing of all login and control paths through the program. The most effective test programs use artificial test data generated by people other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications. The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

.

7.2.2 OUTPUT SCREEN



File Edit Selection View Go Run Terminal Help
CrimePrototype
EXPLORER
CRIMEPROTOTYPE
.reef
.vscode
CrimePrototype
templates
app.py
PROJECTS
PROJECT DEPENDENCIES
app.py
Click here to ask Blackbox to help you code faster
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE SEARCH ERROR
JavaSE-21
Python
Python: app
Connect ObjectScript Share Code Link Explain Code Comment Code Find Bugs Code Chat Blackbox Search Error
Go Live Blackbox 18 Spell bsc-0.65.27 Remote: disabled

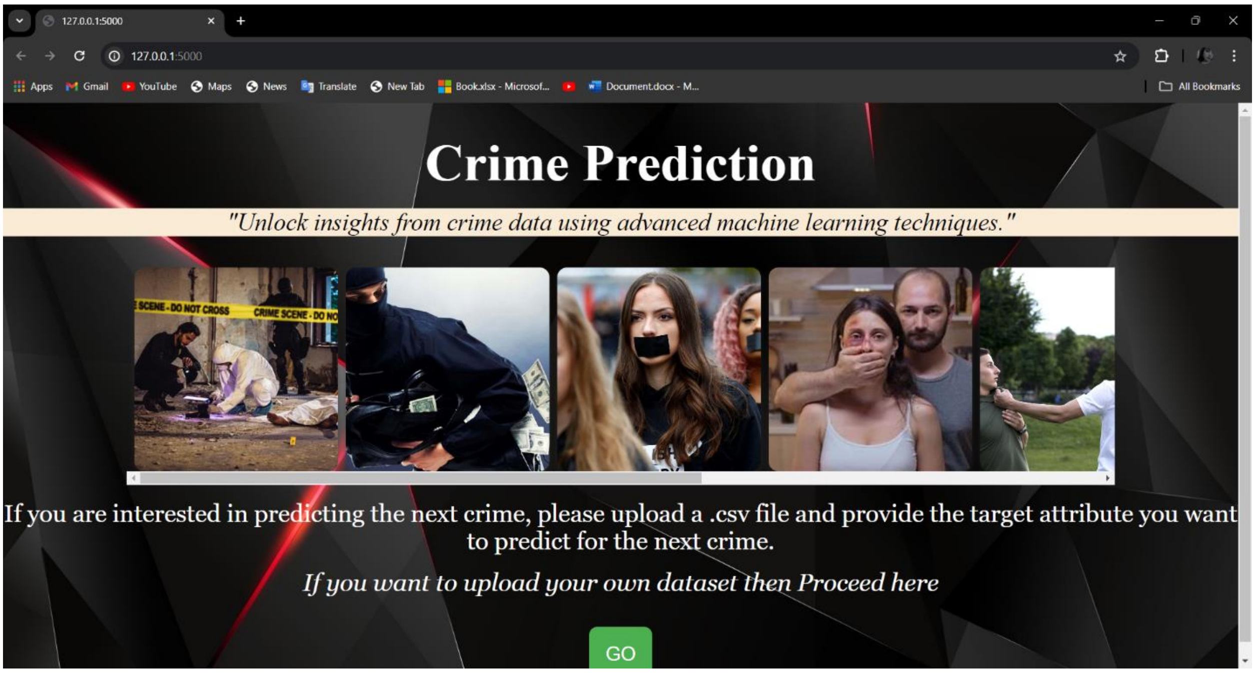
```python
from flask import Flask, render_template, request, redirect, url_for
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
import io
import base64
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from datetime import datetime
import warnings
from warnings import filterwarnings
filterwarnings('ignore')
app = Flask(__name__, static_url_path='/static/styles.css')

@app.route('/')
def home():
    return render_template('home.html')
@app.route('/index')
def index():
    return render_template('index.html')
```

```
SyntaxError: invalid syntax

C:\Users\kudar\Downloads\CrimePrototype>C:/Users/kudar/AppData/Local/Programs/Python/Python311/python.exe c:/Users/kudar/Downloads/CrimePrototype/Cr
imePrototype/app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 117-001-151
```

FIG:7.1 Python terminal



127.0.0.1:5000
Apps Gmail YouTube Maps News Translate New Tab Book.xlsx - Microsof... Document.docx - M...
All Bookmarks
GO

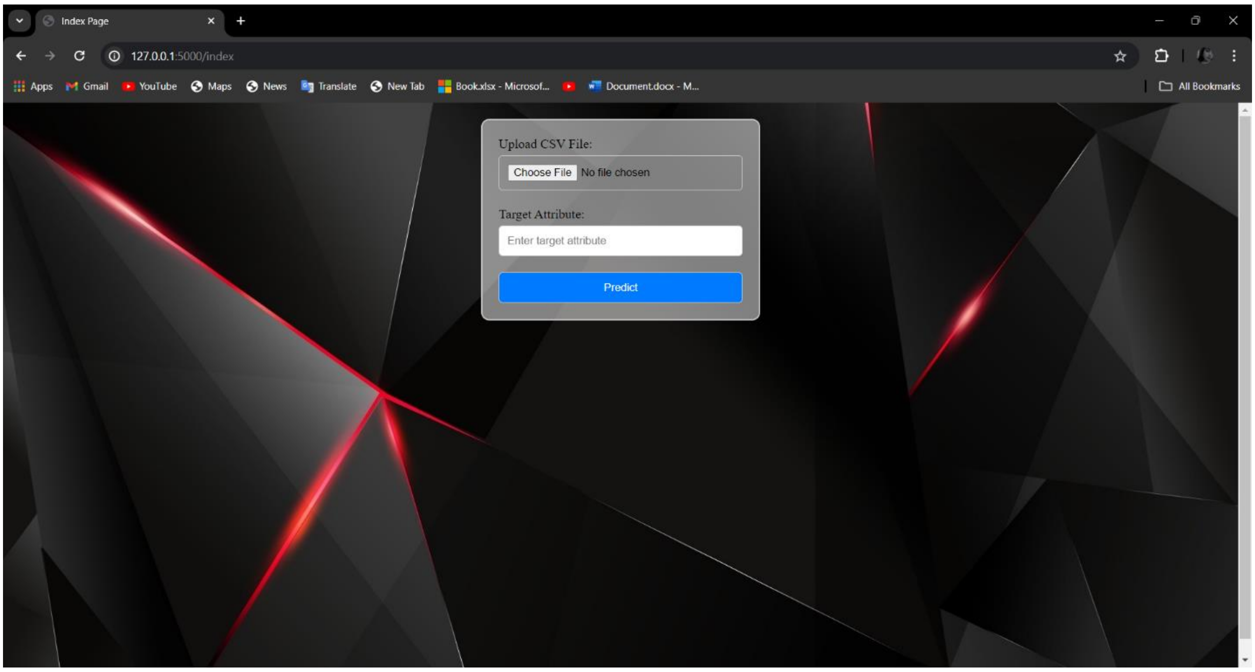
Crime Prediction

"Unlock insights from crime data using advanced machine learning techniques."

If you are interested in predicting the next crime, please upload a .csv file and provide the target attribute you want to predict for the next crime.

If you want to upload your own dataset then Proceed here

FIG:7.2 Home Page



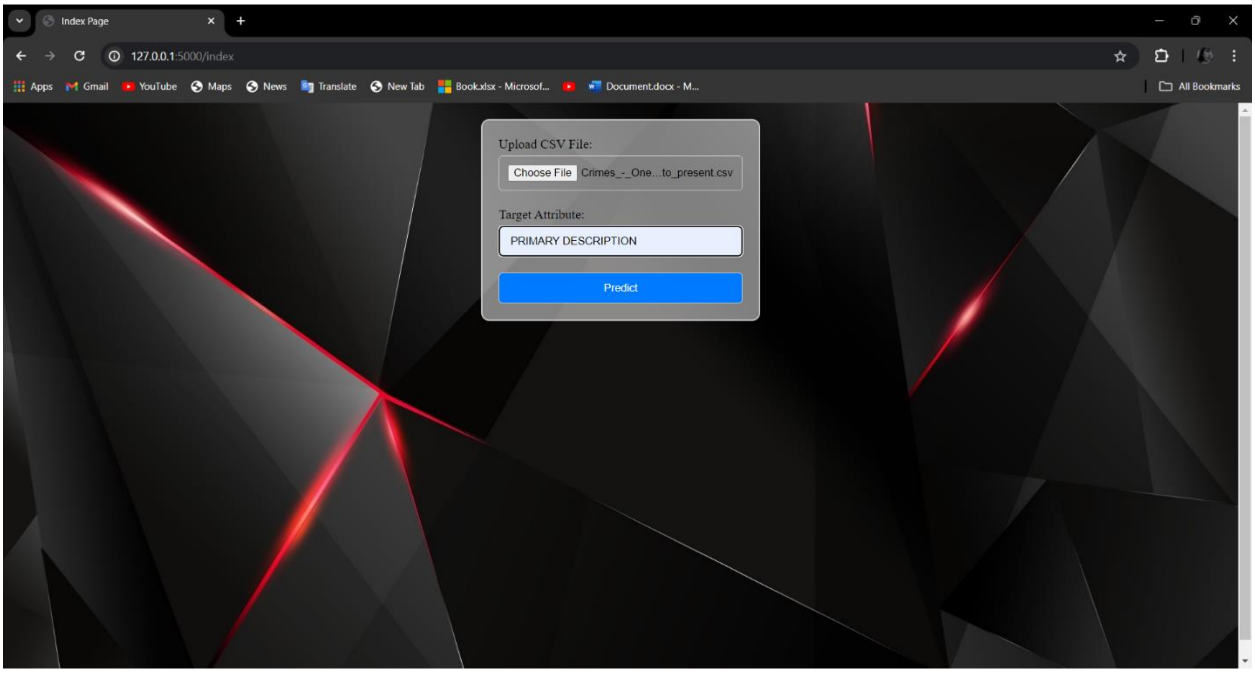
Upload CSV File:

Choose File No file chosen

Target Attribute:

Enter target attribute

FIG:7.3 Index Page



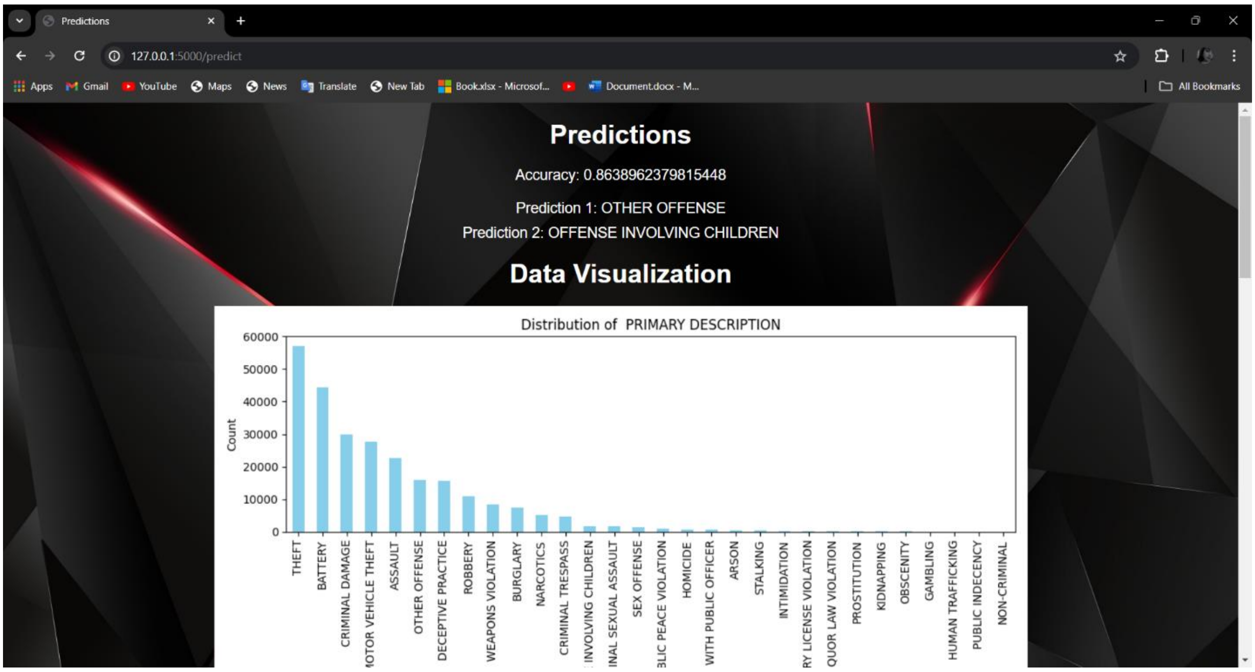
Upload CSV File:

Choose File Crimes_-_One...to_present.csv

Target Attribute:

PRIMARY DESCRIPTION

FIG:7.4 Index Page after uploading dataset and target variable



Predictions

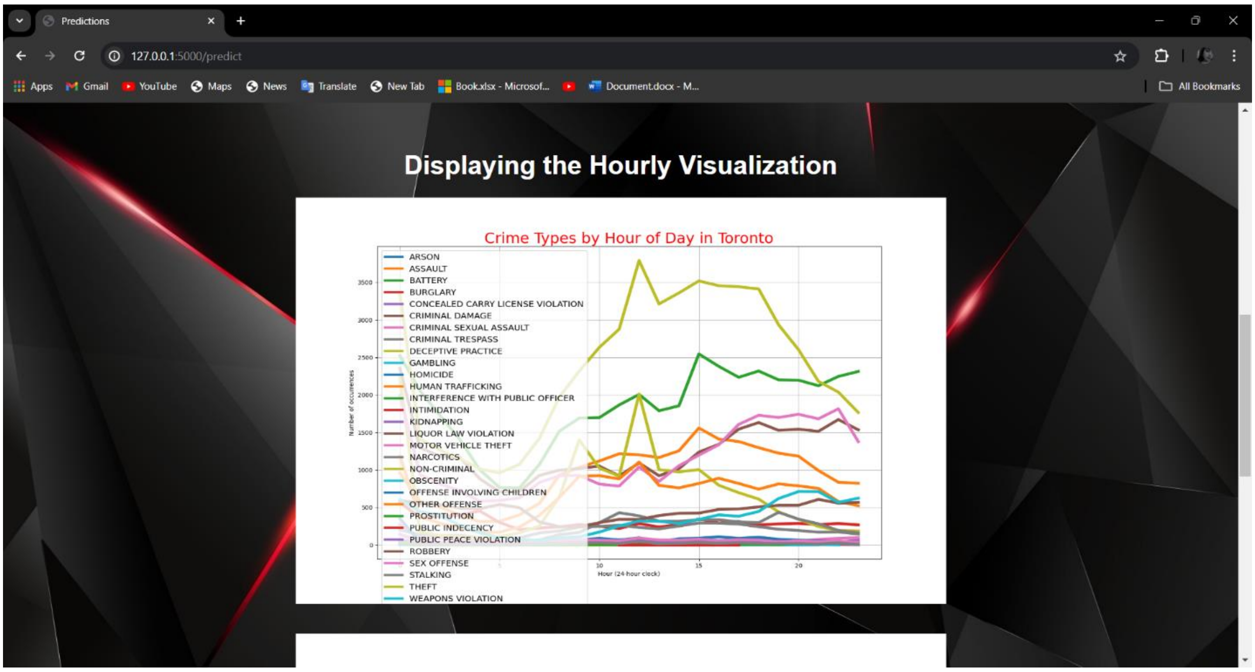
Accuracy: 0.8638962379815448

Prediction 1: OTHER OFFENSE

Prediction 2: OFFENSE INVOLVING CHILDREN

Data Visualization

FIG:7.5 Prediction page which display accuracy, prediction and data visualization



Displaying the Hourly Visualization

FIG:7.6 Displaying the hourly visualization

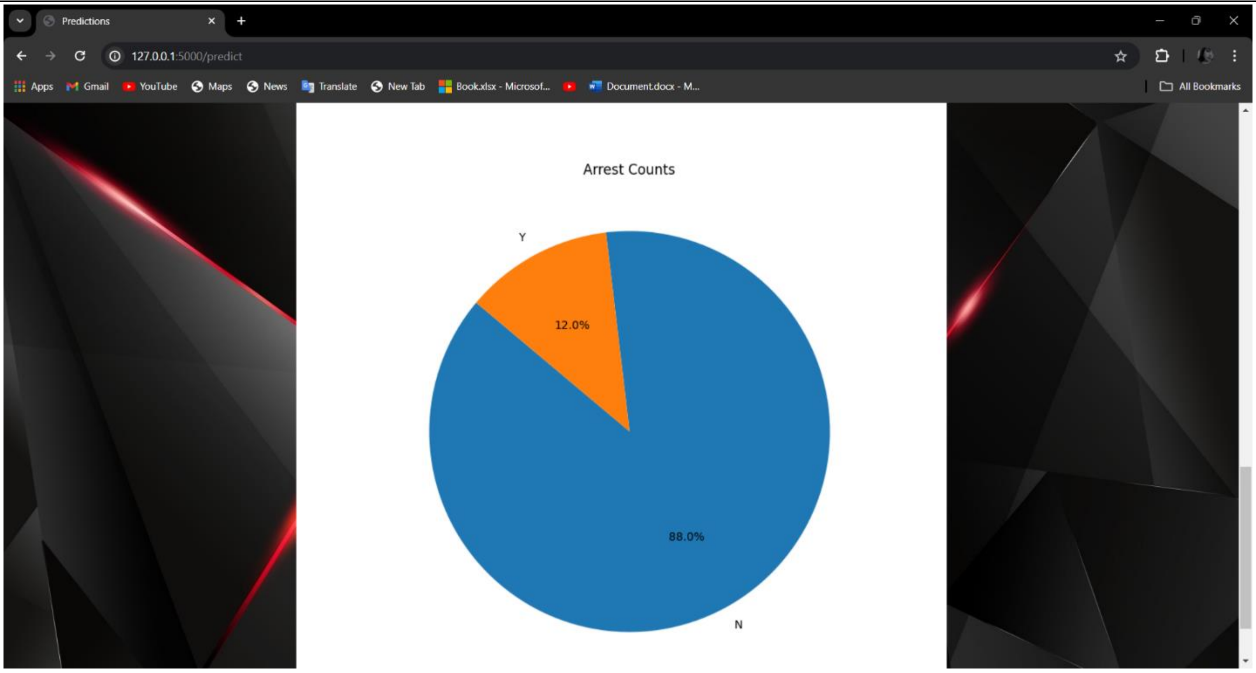


FIG:7.7 Pie char of arrest counts

Chapter 8

CONCLUSION

In this project, we have explored the application of the K-Nearest Neighbors (KNN) algorithm for crime prediction, We have demonstrated the feasibility and effectiveness of using machine learning techniques in this domain. The accuracy of crime prediction models heavily relies on the quality and quantity of available data, as well as the choice of features and parameters in the algorithm. We have overcome the accuracy rate compare to previous machine learning models.

Chapter 9

Future Scope

Future research directions could focus on enhancing the predictive capabilities of crime prediction models by incorporating additional data sources, such as social media activity, weather patterns, and urban infrastructure. Moreover, exploring advanced machine learning techniques, such as deep learning and ensemble methods, may further improve the accuracy and robustness of predictive models in this domain.

REFRENCES

1. McClendon, Lawrence, and Natarajan Meghanathan. "Using machine learning algorithms to analyze crime data." Machine Learning and Applications: An International Journal (MLAIJ) 2.1 (2015): 1-12.

2. Alkesh Bharati, Dr Sarvanaguru RA. K,” Crime Prediction and Analysis Using Machine Learning” in International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 09| September 2018

3. McClendon, L., & Meghanathan, N. (2015). Using machine learning algorithms to analyze crime data. Machine Learning and Applications: An International Journal (MLAIJ), 2(1), 1-12.

4. Chen, Hsinchun, et al. "Crime data mining: a general framework and some examples." computer 37.4 (2004): 50-56.

5. Chen, H., Chungcdvsfdf, W., Xu, J. J., Wangsac, G., Qin, Y., & Chauascas, M. (2004). Crime data mining: a general framework and some examples. computer, 37(4), 50-56.

6. Chen, Hsinchun, Wingyan Chung, Jennifer Jie Xu, Gang Wang, Yi Qin, and Michael Chau."Crime data mining: a general framework and some examples." computer 37, no. 4 (2004): 50-56.

7. Chen, H., Chungsasda, W., Xu, J.J., Wang, G., Qin, Y. and Chau, M., 2004. Crime data mining: a general framework and some examples. computer, 37(4), pp.50-56.

8. Sathyadevan, Shiju. "Crime analysis and prediction using data mining." 2014 First International Conference on Networks & Soft Computing (ICNSC2014). IEEE, 2014.

9. Sathyadevan, S. (2014, August). Crime analysis and prediction using data mining. In 2014 First International Conference on Networks & Soft Computing (ICNSC2014) (pp. 406-412). IEEE.

10. Sathyadevan, Shiju. "Crime analysis and prediction using data mining." In 2014 First International Conference on Networks & Soft Computing (ICNSC2014), pp. 406-412. IEEE, 2014.

11. Sathyadevan, S., 2014, August. Crime analysis and prediction using data mining. In 2014 First International Conference on Networks & Soft Computing (ICNSC2014) (pp. 406-412). IEEE.

12. Sathyadevan S. Crime analysis and prediction using data mining. In2014 First International Conference on Networks & Soft Computing (ICNSC2014) 2014 Aug 19 (pp. 406-412). IEEE.

13. Nath, Shyam Varan. "Crime pattern detection using data mining." 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops. IEEE, 2006.

14. Nath, S. V. (2006, December). Crime pattern detection using data mining. In 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops (pp. 41-44). IEEE.

15. Nath, Shyam Varan. "Crime pattern detection using data mining." In 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, pp. 41-44. IEEE, 2006.

16. Nath, S.V., 2006, December. Crime pattern detection using data mining. In 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops (pp.41-44). IEEE.

17. Nath SV. Crime pattern detection using data mining. In2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops 2006 Dec 18 (pp.41-44). IEEE.

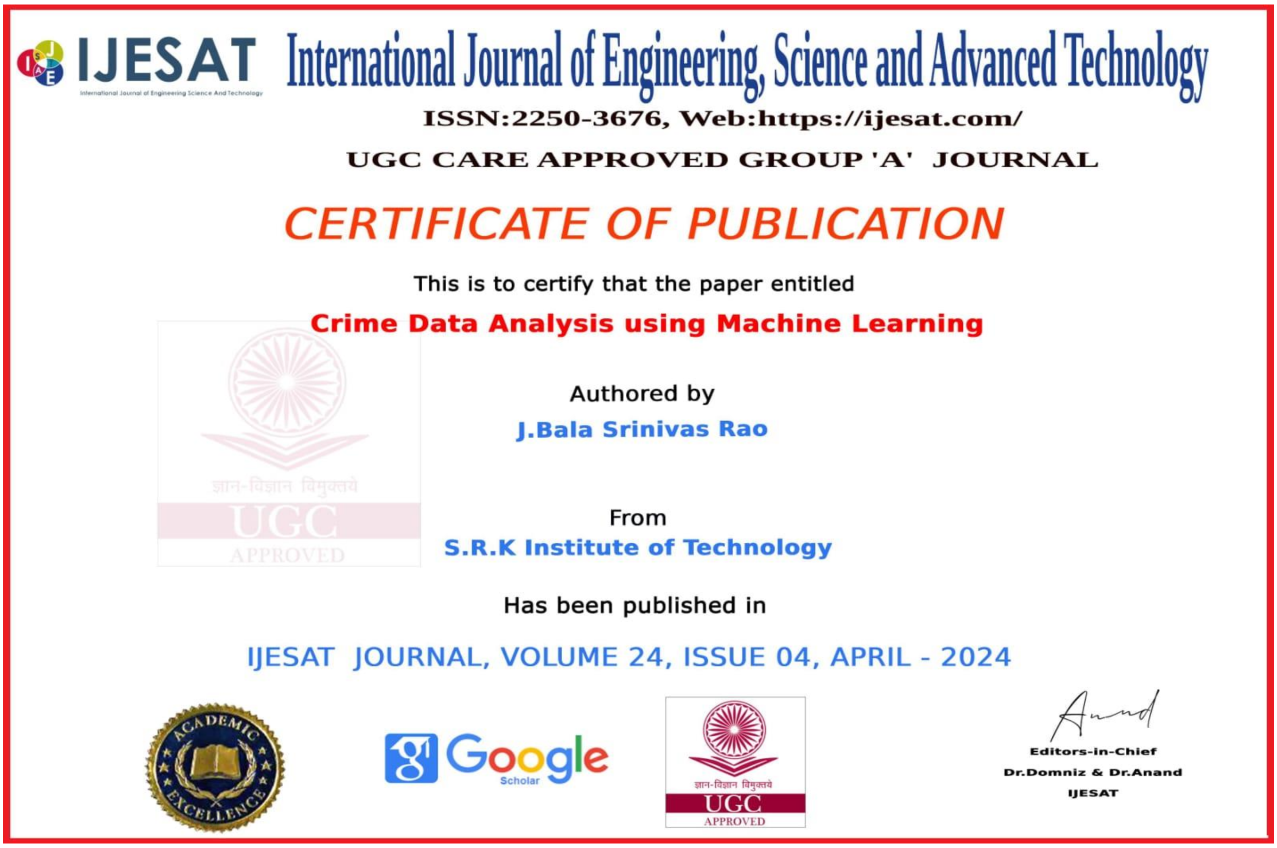
18. Wang, Tong, et al. "Learning to detect patterns of crime." Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, 2013.

19. Wang, T., Rudin, C., Wagner, D., & Sevieri, R. (2013, September). Learning to detect patterns of crime. In Joint European conference on machine learning and knowledge discovery in databases (pp. 515-530). Springer, Berlin, Heidelberg.

20. Wang, Tong, et al. "Learning to detect patterns of crime." Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, 2013.

PUBLISHED PAPER REPORT

GUIDE CERTIFICATE:



IJESAT International Journal of Engineering, Science and Advanced Technology

ISSN:2250-3676, Web:https://ijesat.com/

UGC CARE APPROVED GROUP 'A' JOURNAL

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

Crime Data Analysis using Machine Learning

Authored by

J.Bala Srinivas Rao

From

S.R.K Institute of Technology

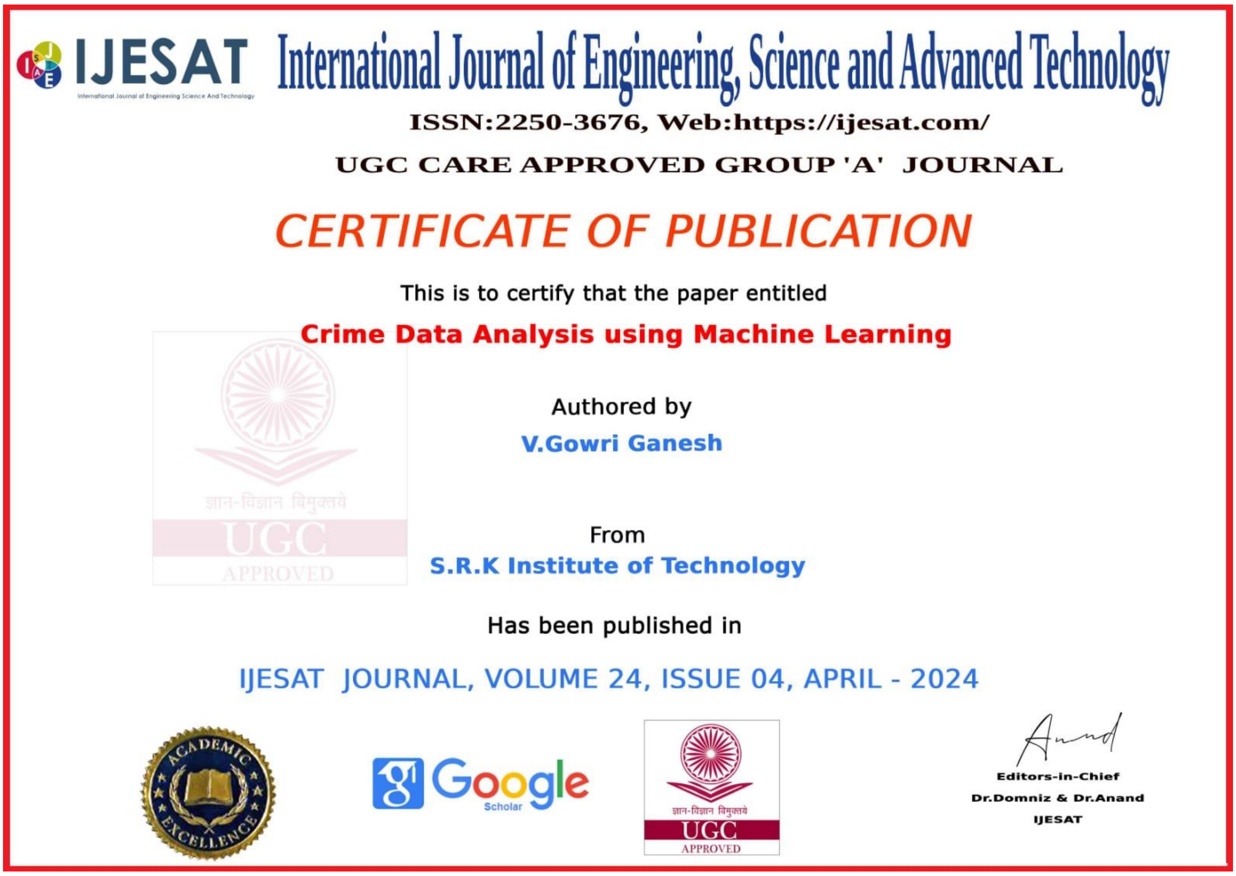
Has been published in

IJESAT JOURNAL, VOLUME 24, ISSUE 04, APRIL - 2024

Anand

Editors-in-Chief
Dr.Domniz & Dr.Anand
IJESAT

TEAM LEADER CERTIFICATE:



IJESAT

International Journal of Engineering Science And Technology

International Journal of Engineering, Science and Advanced Technology

ISSN:2250-3676, Web:https://ijesat.com/

UGC CARE APPROVED GROUP 'A' JOURNAL

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

Crime Data Analysis using Machine Learning

Authored by

V.Gowri Ganesh

From

S.R.K Institute of Technology

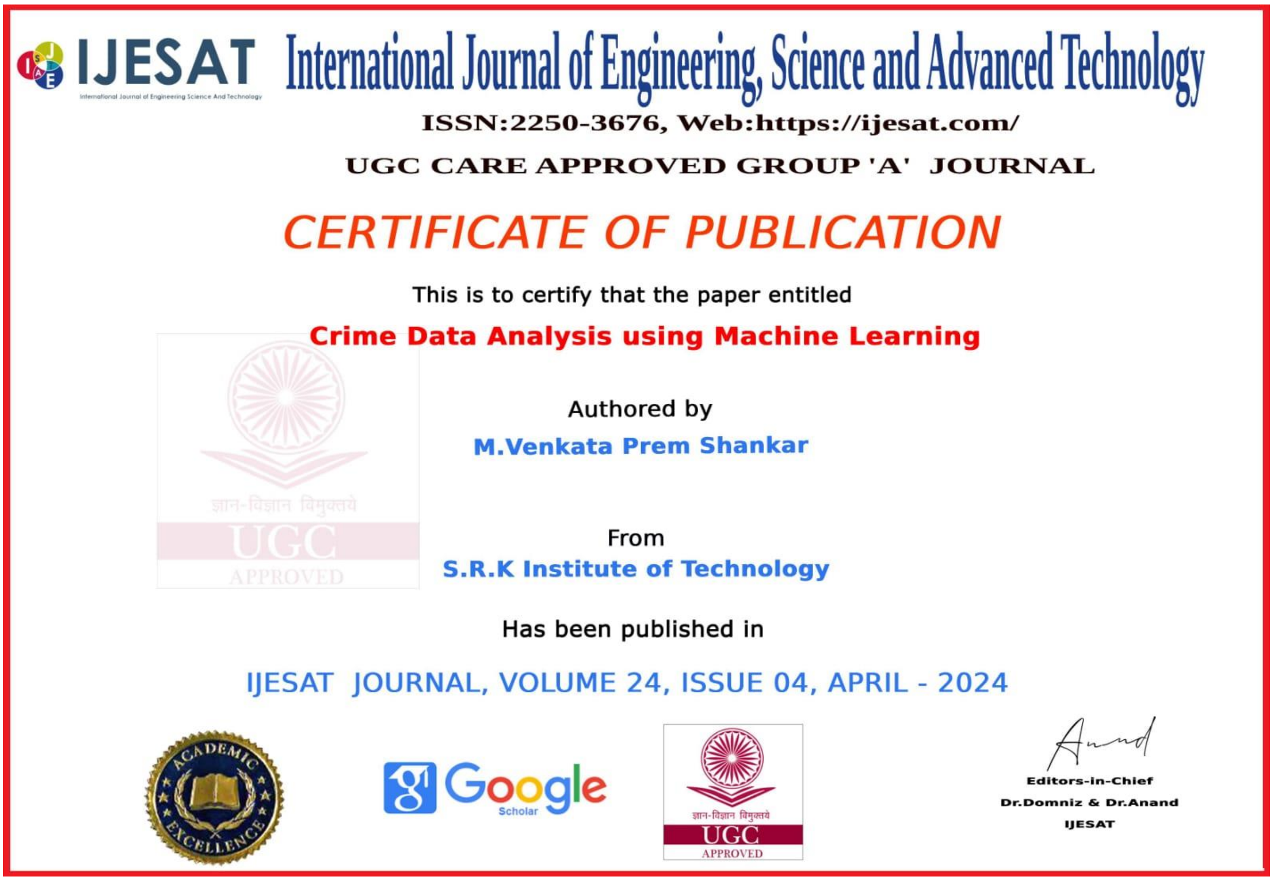
Has been published in

IJESAT JOURNAL, VOLUME 24, ISSUE 04, APRIL - 2024

Anand

Editors-in-Chief
Dr.Domniz & Dr.Anand
IJESAT

TEAM MEMBERS CERTIFICATES:



IJESAT International Journal of Engineering, Science and Advanced Technology

International Journal of Engineering Science And Technology

ISSN:2250-3676, Web:https://ijesat.com/

UGC CARE APPROVED GROUP 'A' JOURNAL

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

Crime Data Analysis using Machine Learning

Authored by

M.Venkata Prem Shankar

From

S.R.K Institute of Technology

Has been published in

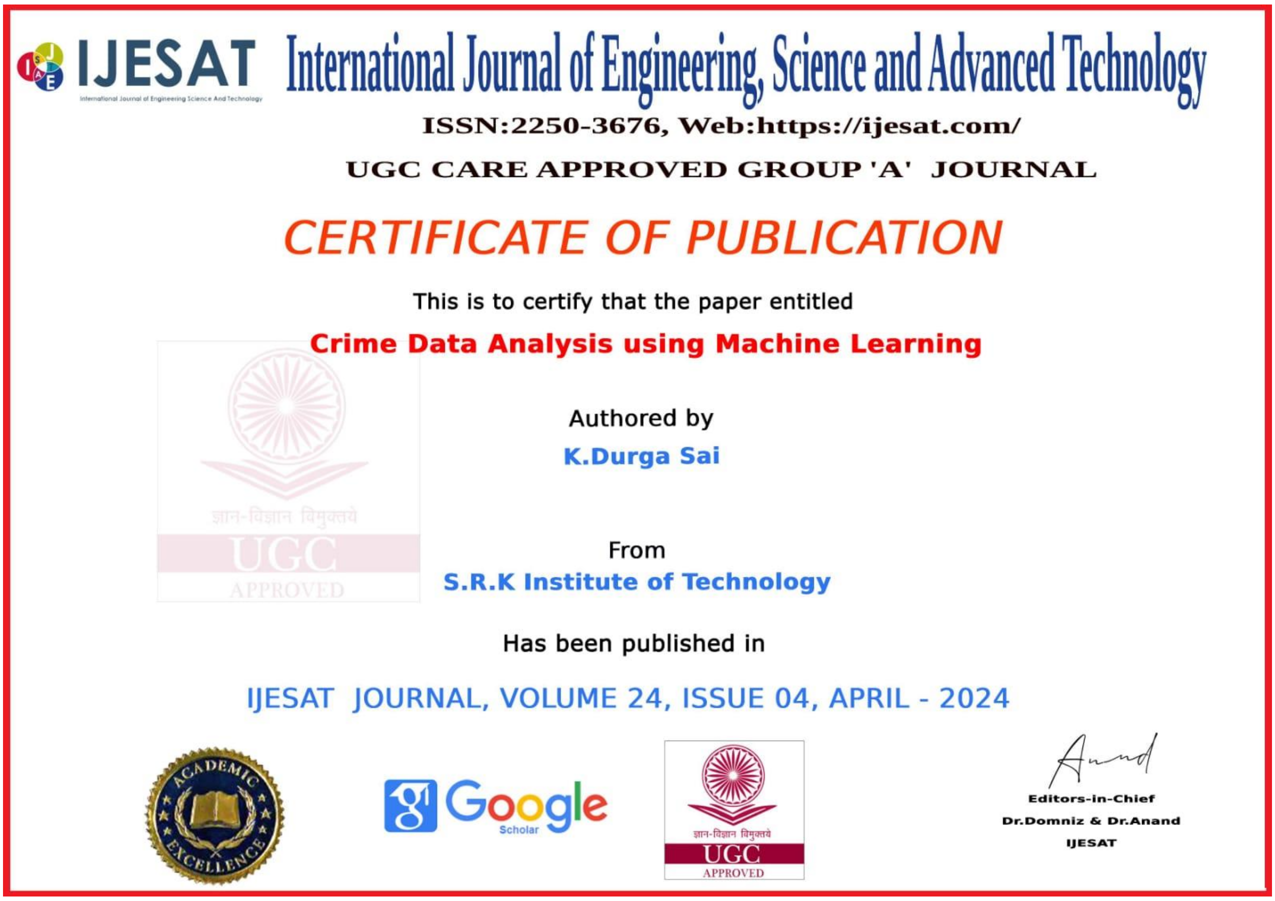
IJESAT JOURNAL, VOLUME 24, ISSUE 04, APRIL - 2024

ACADEMIC EXCELLENCE

Google Scholar

ज्ञान-विज्ञान विमुक्तये UGC APPROVED

Editors-in-Chief
Dr.Domniz & Dr.Anand
IJESAT



IJESAT International Journal of Engineering Science And Technology

International Journal of Engineering, Science and Advanced Technology

ISSN:2250-3676, Web:https://ijesat.com/

UGC CARE APPROVED GROUP 'A' JOURNAL

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

Crime Data Analysis using Machine Learning

Authored by

K.Durga Sai

From

S.R.K Institute of Technology

Has been published in

IJESAT JOURNAL, VOLUME 24, ISSUE 04, APRIL - 2024

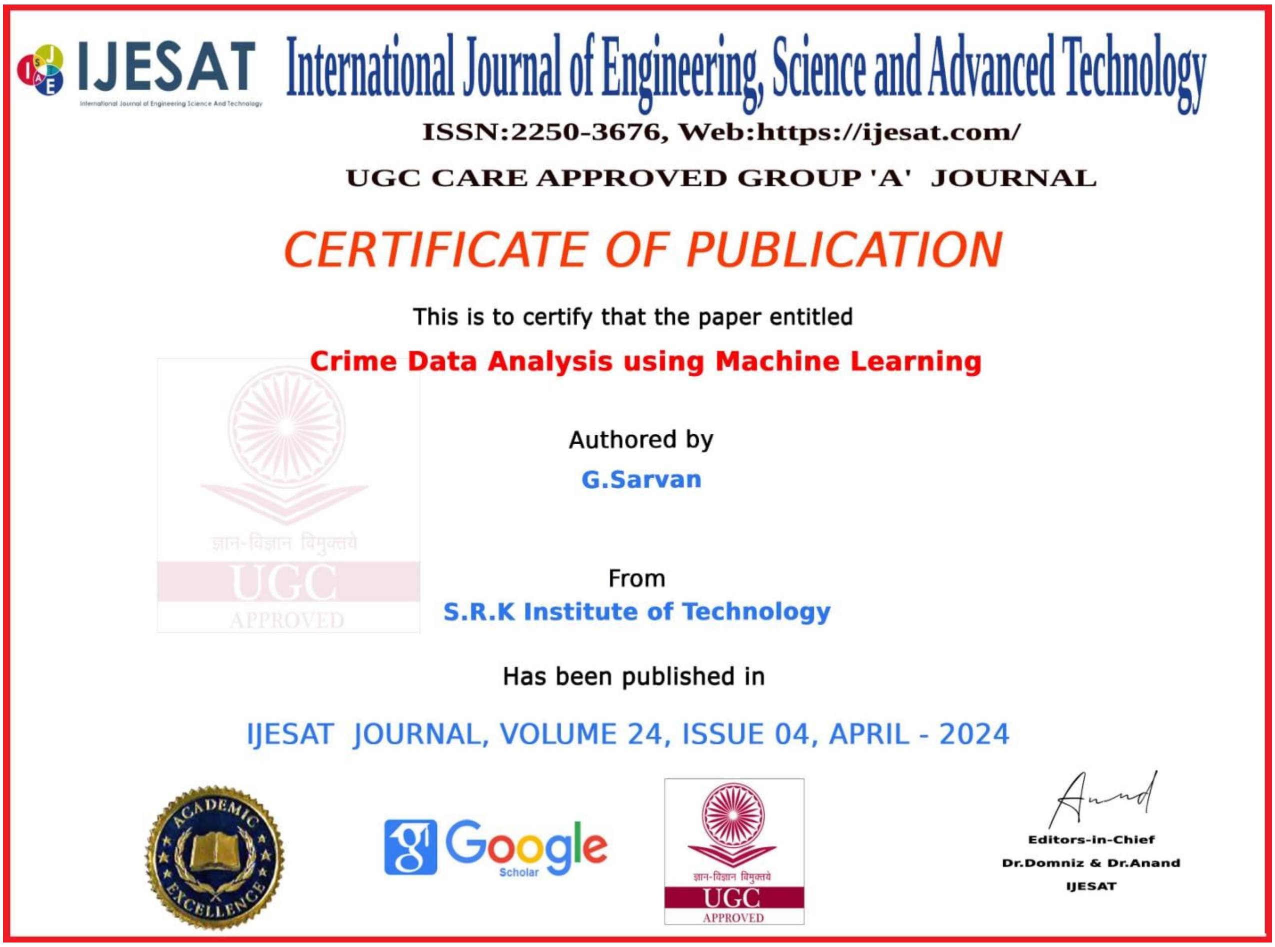
ACADEMIC EXCELLENCE

Google Scholar

ज्ञान-विज्ञान विमुक्तये
UGC
APPROVED

Anand

Editors-in-Chief
Dr.Domniz & Dr.Anand
IJESAT



IJESAT International Journal of Engineering, Science and Advanced Technology

ISSN:2250-3676, Web:https://ijesat.com/

UGC CARE APPROVED GROUP 'A' JOURNAL

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

Crime Data Analysis using Machine Learning

Authored by

G.Sarvan

From

S.R.K Institute of Technology

Has been published in

IJESAT JOURNAL, VOLUME 24, ISSUE 04, APRIL - 2024

Anand

Editors-in-Chief
Dr.Domniz & Dr.Anand
IJESAT

Crime Data Analysis using Machine Learning

J.Bala Srinivas Rao[1], V.Gowri Ganesh[2], M.Venkata Prem Shankar[3] ,G.Sarvan[4], K.Durga Sai[5]

[1]Assisant Professor, Department of CSE-Artificial Intelligence and Machine Learning , S.R.K Institute of Technology, NTR, Andhra Pradesh, India, voonnagowriganesh@gmail.com.

[2]*student, Department of CSE-Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India*

[3]*student, Department of CSE- CSE-Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India*

[4]*student, Department of CSE- Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India*

[5]*student, Department of CSE- Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India*

Abstract**—** Criminal cases are rapidly increasing in our society day to day. These leading to backlog of pending cases. It is important to control the increasing crimes else it becomes tough to handle for Law Enforcement Agencies, they store the information of every crime after happening because there might be a chance of Pattern behind the occurrence of every crime so in order to control the crimes we are going to create a Machine Learning Model to predict the Crime pattern by training the model through K Nearest Neighbour (KNN) Algorithm to get more predictions accurately than existed Algorithms like Naïve Bayes, Decision Tree etc… Already the people worked on KNN algorithm to train model they used the dataset and get prediction rate of 75% but now we are working on same algorithm to get more than 85%. Also we used to work on this approach to reduce the code complexity. We would show the chance of occurrence of next two crime types based on our dataset. We collected the dataset from data.govern.in at the period of 2023 Jan - 2024 Jan.

Keywords**—** KNN, Machine Learning, Crime Prediction, Data Visualization, Accuracy, Pattern Recognition.

I.INTRODUCTION

Crimes are harmful actions that lead to threats to human lives. Crimes might be Robbery, Murder, Rape, Women trafficking, etc. As the population increases the rate of crime also increases day by day. The increasing cases lead to a backlog of pending cases to the police department, The crime activities have increased at a faster rate and it is the responsibility of the police department to control and reduce the crime activities. the department tries to solve the cases according to the evidence they got but in major cases, it is not as much possible to solve and decrease the crime rate as they think.

This analysis leads us to research the crimes to make them complex and free for solving the cases. The main thing here we are going to work on is predicting the occurrence of the next crime. It might be helpful to the Law Enforcement agencies and police departments to control and be aware of the respective situation. It will only be possible by collecting the previous information. So, we get the information stored in dataset format in which the dataset contains the relative features like crime type, place, time, arrest or not, victims, and whether the case is solved or not, etc… We could extract the dataset from official site data.govern.in. The prediction of the occurrence of crime can happen by working with a machine learning model and one optimal algorithm, here we are going to work with K Nearest Neighbour which is well-suited algorithm for both classification and regression and can also get good prediction accuracy. Visualization of occurrence of crimes are well-known thing to the normal people hence we could also implement the work with visual graphs.

II. EXISTING SYSTEM

In existing Systems, they used Naïve Bayes algorithm which is a supervised learning algorithm which is used for classification. Mainly used for text classification based on the training dataset. Naïve Bayes algorithm assumes all features were independent to each other. It depends on the conditional probability.

Formula :

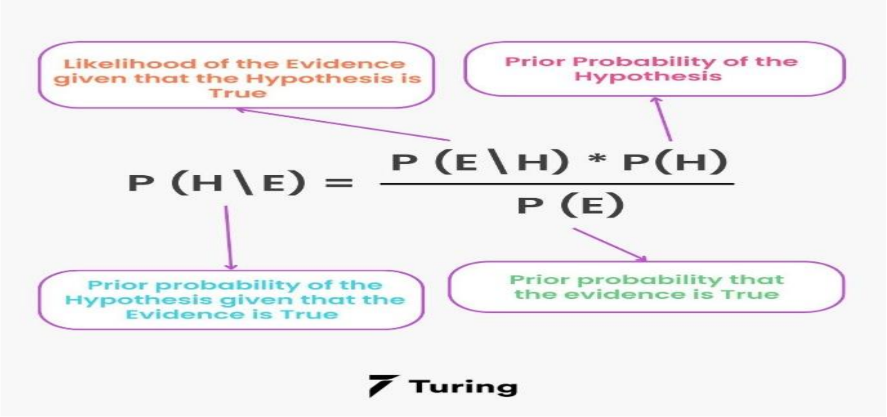

Likelihood of the Evidence given that the Hypothesis is True
Prior Probability of the Hypothesis
P (H\E) = P (E\H) * P(H) / P (E)
Prior probability of the Hypothesis given that the Evidence is True
Prior probability that the evidence is True
Turing

Fig. 1 Naïve Bayes formula.

Disadvantages:

- Shows lower performance compared to other classification models.
- Require large Data records to achieve a good accuracy result.
- Features are independent to each other therefore it results in low accuracy.

III.CONCEPTS OF PROPOSED SYSTEM

A. Predictive Modelling

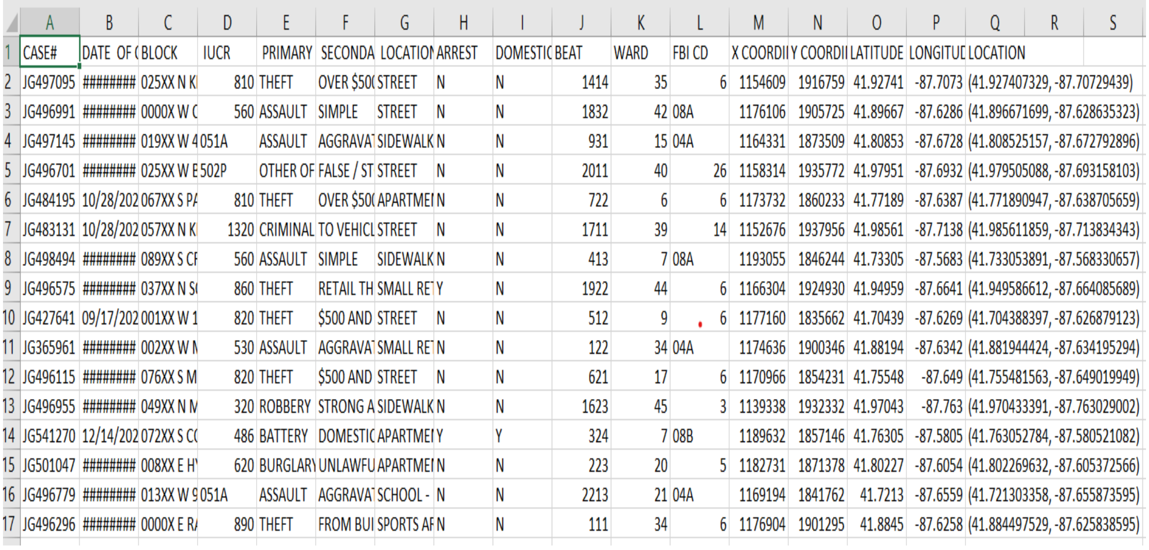
The concept that we have that is predictive modelling that is any model that we want to build is used to predict the results in order that based on how it had trained. In the process that includes machine learning algorithm that trained from fed dataset. The modelling has divided into two types classification and model regression which describes the analysis of tremendous research between the trends and variables. When it becomes to regression tasks it allows you to assign the class labels to different classes that is assumes a group or a class named as class A, we can simply state to predict that whether a boy can enjoy the sport under different kind of weather conditions.

In the other hand Pattern classification can divided into two parts those are Supervised Machine Learning model and Unsupervised machine learning model. In supervised machine learning model, the dataset can well know with its features and data with also what type of data we are feeding and trains to the model to get accurate predictions can be made for unknown data. when we come and talking about Unsupervised learning the scenario is quite opposite to supervised learning model.

B. Types of predictive modelling

Generally, we all aware on decision tree which emits the possible number of outcomes as graph or tree shaped liked structured, which used as a classification algorithm. It is a chance to show the algorithm. In the phenomenon which we get possible number of predictive outcomes as the result just like the algorithms like decision tree. Here the problem's features assign to the actual algorithm class labels to make a line of bond to construct the algorithmic approach, class labels can get from the known set. Naïve Bayes algorithm which is very closest approach algorithm uses probabilistic classifier based the applied bayes theorem with independent factors among the classes. We can also state the Naïve Bayes is family of probabilistic classifier. Linear Regression more aware of a supervised machine learning algorithm approach used to maps the datapoints to the most optimized linear functions. It involves only one dependent and one independent variable. logistic regression, it is a regression model where the dependent variable is categorical, or we can say binary.

1.Dataset:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	CASE#	DATE OF (	BLOCK	IUCR	PRIMARY	SECONDA	LOCATION	ARREST	DOMESTIC	BEAT	WARD	FBI CD	X COORDI	Y COORDI	LATITUDE	LONGITUD	LOCATION		
2	JG497095	########	025XX N K	810	THEFT	OVER $50(	STREET	N	N	1414	35	6	1154609	1916759	41.92741	-87.7073	(41.927407329, -87.70729439)		
3	JG496991	########	0000X W (	560	ASSAULT	SIMPLE	STREET	N	N	1832	42	08A	1176106	1905725	41.89667	-87.6286	(41.896671699, -87.628635323)		
4	JG497145	########	019XX W 4	051A	ASSAULT	AGGRAVAT	SIDEWALK	N	N	931	15	04A	1164331	1873509	41.80853	-87.6728	(41.808525157, -87.672792896)		
5	JG496701	########	025XX W E	502P	OTHER OF	FALSE / ST	STREET	N	N	2011	40	26	1158314	1935772	41.97951	-87.6932	(41.979505088, -87.693158103)		
6	JG484195	10/28/202	067XX S P/	810	THEFT	OVER $50(	APARTMEI	N	N	722	6	6	1173732	1860233	41.77189	-87.6387	(41.771890947, -87.638705659)		
7	JG483131	10/28/202	057XX N K	1320	CRIMINAL	TO VEHICL	STREET	N	N	1711	39	14	1152676	1937956	41.98561	-87.7138	(41.985611859, -87.713834343)		
8	JG498494	########	089XX S CI	560	ASSAULT	SIMPLE	SIDEWALK	N	N	413	7	08A	1193055	1846244	41.73305	-87.5683	(41.733053891, -87.568330657)		
9	JG496575	########	037XX N S(	860	THEFT	RETAIL TH	SMALL RE	Y	N	1922	44	6	1166304	1924930	41.94959	-87.6641	(41.949586612, -87.664085689)		
10	JG427641	09/17/202	001XX W 1	820	THEFT	$500 AND	STREET	N	N	512	9	6	1177160	1835662	41.70439	-87.6269	(41.704388397, -87.626879123)		
11	JG365961	########	002XX W N	530	ASSAULT	AGGRAVAT	SMALL RE	N	N	122	34	04A	1174636	1900346	41.88194	-87.6342	(41.881944424, -87.634195294)		
12	JG496115	########	076XX S M	820	THEFT	$500 AND	STREET	N	N	621	17	6	1170966	1854231	41.75548	-87.649	(41.755481563, -87.649019949)		
13	JG496955	########	049XX N M	320	ROBBERY	STRONG A	SIDEWALK	N	N	1623	45	3	1139338	1932332	41.97043	-87.763	(41.970433391, -87.763029002)		
14	JG541270	12/14/202	072XX S C(	486	BATTERY	DOMESTIC	APARTMEI	Y	Y	324	7	08B	1189632	1857146	41.76305	-87.5805	(41.763052784, -87.580521082)		
15	JG501047	########	008XX E H	620	BURGLARY	UNLAWFU	APARTMEI	N	N	223	20	5	1182731	1871378	41.80227	-87.6054	(41.802269632, -87.605372566)		
16	JG496779	########	013XX W 9	051A	ASSAULT	AGGRAVAT	SCHOOL -	N	N	2213	21	04A	1169194	1841762	41.7213	-87.6559	(41.721303358, -87.655873595)		
17	JG496296	########	0000X E R/	890	THEFT	FROM BUI	SPORTS AF	N	N	111	34	6	1176904	1901295	41.8845	-87.6258	(41.884497529, -87.625838595)		

Fig. 2 Dataset Image.

C. *Data Preprocessing*

The information involves any null values are unnecessary duplicates that might cause of misleads to the Target. It also affects the work accuracy or algorithm accuracy, So the process involves in removing null values and replacing with respective values in it, simply we can say that handling of null values and duplicates. The process can also move forward with some probabilistic approaches like mean, median and mode. The main mechanism involves in three steps they are Cleaning, Sampling and Formatting. By using these steps in python as preprocessing we reduce the running rate get optimal time of run.

D. *Functional Diagram of Proposed Work*

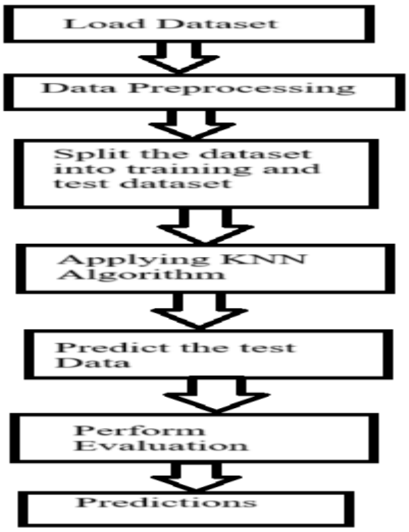


Fig. 3 Functional Diagram of Proposed Work

E. Prepare Data

- Have to collect the data from right sources without lack of necessary information.
- Must of data cleaning
- Study and covert or transform the variables
- We can transform the variables by using any approach from the below.

1) Standardization or normalization.
2) Missing value operation.

F. Random Sampling(Train or Test)

1.Training sample :

Training data is used to train the model for future accuracy prediction with information of 70% to 80% data

2.Test sample :

Testing data is used to test and validate the data that we were we fed to the model that is to check whether our model works well on train data or not. The test data maybe about 20% to 30%.

G. Model Selection

According the problem we have, we have to choose the appropriate model or approach to deal the situation is necessary. Based on the problem we have to choose either one or combination of modelling techniques that we have such as,

Decision Tree
Logistic Regression
Super Vector Machine (SVM)
KNN classification
Bayesian Methods
Random Forest

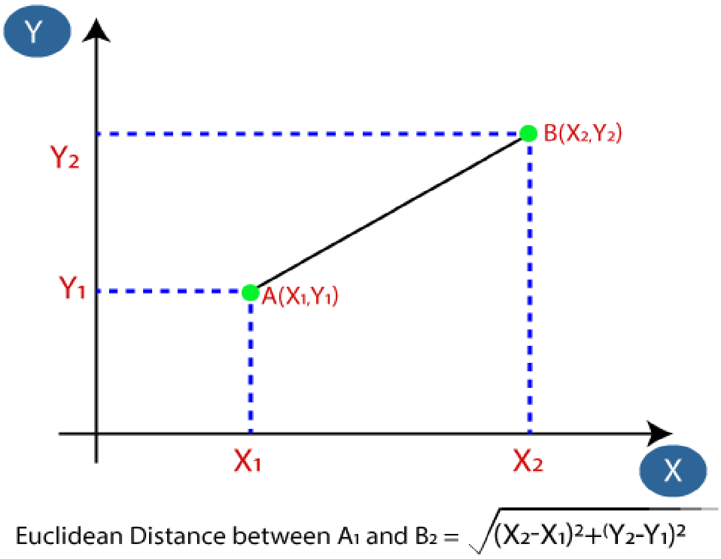

Y
Y_2
$B(X_2,Y_2)$
Y_1
$A(X_1,Y_1)$
X_1
X_2
X
Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Fig. 4 Proposed KNN Algorithm Formula

***Advantages of proposed algorithm*:**

- Features are dependent which able to give good accuracy.
- It is very simple.
- Able to work even with large Datasets.
- Easy to implement.
- New Data can be added seamlessly.

H. Build/Develop/Train the model

- Validating all possibilities of picked algorithm.

- Based on the available of data it is necessary to train the model sufficiently.
- Validating the model performance like Error and Accuracy.

I. *Validate or test model*

- By using test data we have to validate test accuracy and final the score.
- Checking model performance that is model accuracy

IV.IMPLEMENTATION

We have collected the dataset from officially at data.govern.in that we are using in our current project. This is information is maintained that is updated with every change by Chicago police department.

working on the project is followed by several steps they are

A. *Collection of Information*

We collected the dataset from Data.govern.in in .csv extension.

Our dataset name is Crimes_-_One_year_prior_to_present dataset.

B. *Data preprocessing*

Our dataset consists of --- entries.First we have converted all attributes into numerical datatype using label encoder and then replaced all Null Values using median Strategy.

C. *Feature selection*

Generally feature selection plays a crustial role that is it used to be build the model. The attributes which are used for feature selection are Block,Location,case,X coordinate , Y coordinate, Latitude , Longitude.

D. *Building and Training the model*

Location and --- attributes are used for the training after feature selection and then the dataset is classified into xtrain, ytrain and xtest, ytest. Sklearn is used to import the model alogorithms[fit(xtrain, ytrain)].

E. *Prediction*

model.predict(xtest) is used to done the prediction after done the all previous process and build the model. accuracy_score is used to calculate the accuracy which is acutally imported from the metrics.accuracy_score(ytest, predicted). This is the process we generally used in prediction process.

F. *Visualization*

We used sklearn to import the matplotlib library for visualization. We represented the crime analysis in many ways like plotting graphs and represented in pie charts.

G. *Results and Discussion*

We can obtain the results after undergoing into many processes with many functions that are through machine learning.

V.DATA VISUALIZATION

Crime visualization totally allowed to show the dataset analysis in a visualized format like in way that a normal user can easily go through it. In detail way to plot the graphs or make them to understand by bar graphs etc… The analysis can be done through.

- In a period of time the number of crimes is might committed.
- By taking a city to observe the number of crimes overall crime types
- Ratio of taking them in custody that is the ratio of arrest in police records.
- By considering the different locations observing the committed crimes.
- Details of crimes that are happen majorly in city.

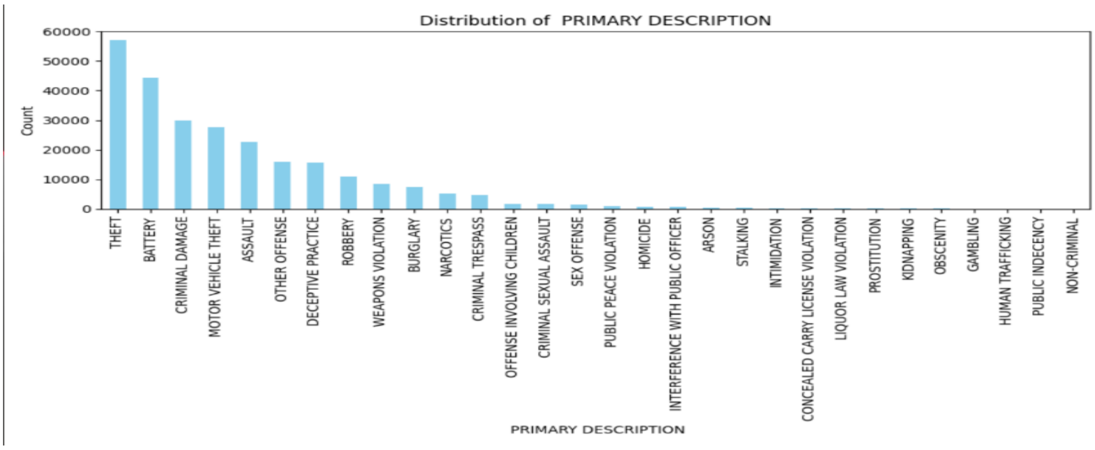


Fig.5 Major Crime Indicator

By analyzing the previous crimes we representing in the form of a bar graph which crime indicates mostly. Here Theft is the major crime indicator, secondary is the Battery Indicator etc… Here we represented x-axis as crime names and y-axis as crimes count. The bar graph represents the major crimes to minor crimes from left to right.

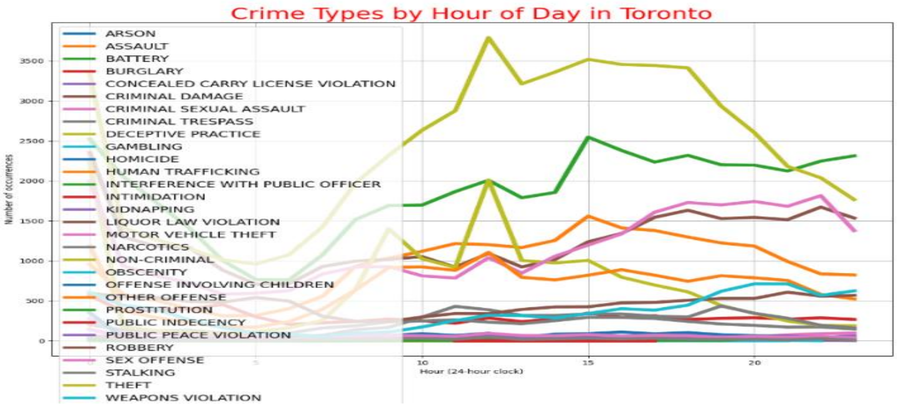


Fig.6 Crime Types by Hours

Here we have used line graph, In one graph we combined all lines into one graph. Here each line represents one type of crime, each line represents the crime and it represents the time on that day of occurrence which is in 24-hr format.

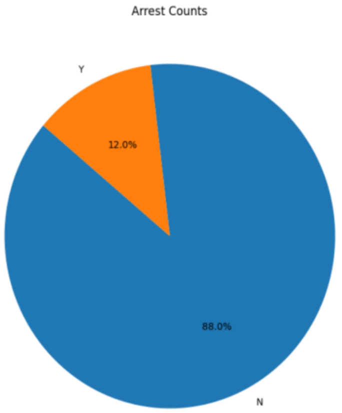
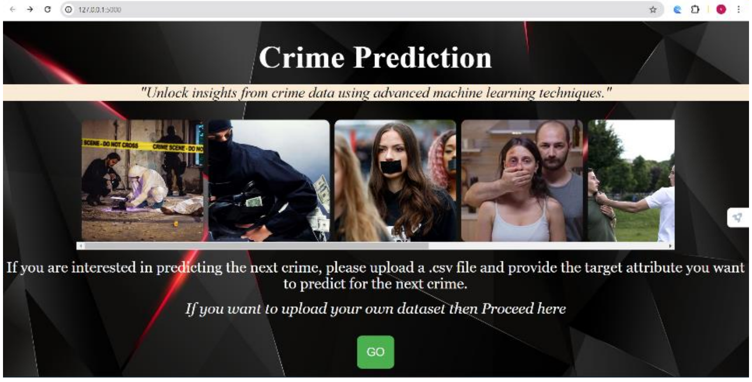


Fig.7 Pie Chart representing the Arrest counts

By using Pie chart we have represented the arrest counts.If we see in past 2023 Jan-2024 Jan only 12% cases accused has been arrest remaining 88% cases are still in pending.

VI.SAMPLE SCREENSHOTS



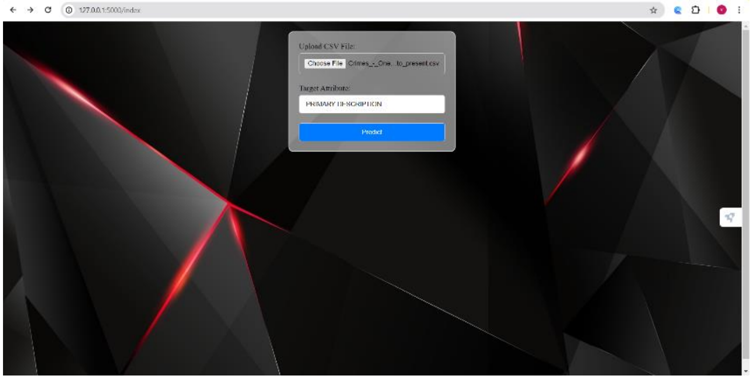
Crime Prediction

"Unlock insights from crime data using advanced machine learning techniques."

If you are interested in predicting the next crime, please upload a .csv file and provide the target attribute you want to predict for the next crime.

If you want to upload your own dataset then Proceed here

Fig.8 Home Page



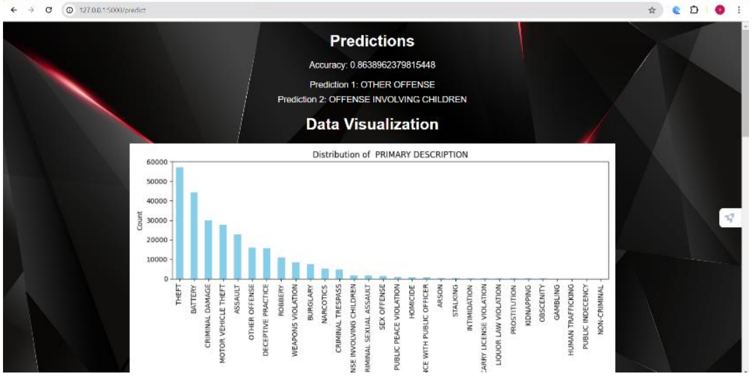
Upload CSV File:

Choose File Crimes_-_One...to_present.csv

Target Attribute:

PRIMARY DESCRIPTION

Fig.9 Index Page



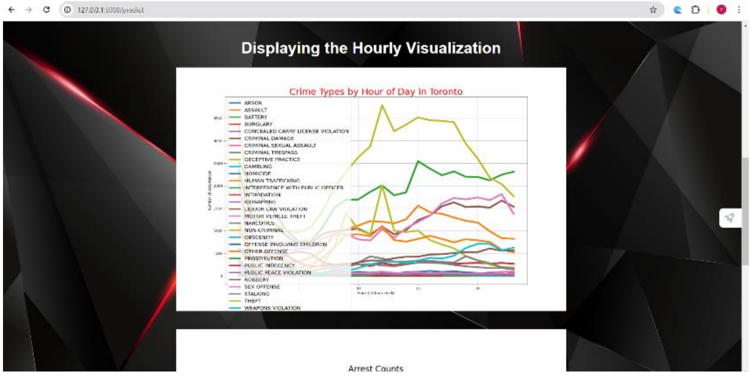
Predictions

Accuracy: 0.8638962379815448

Prediction 1: OTHER OFFENSE
Prediction 2: OFFENSE INVOLVING CHILDREN

Data Visualization

Fig.10(A) Output Page



Displaying the Hourly Visualization

Fig.10(B) Output Page

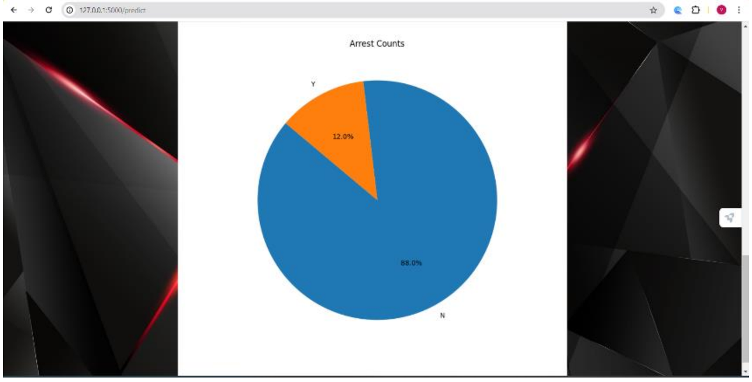


Fig.10(C) Output Page

VII.CONCLUSION

From the entire work we can conclude that our main agenda is to predict the occurrence of crime, it will be possible by know the location of where it occur. The entire process can easily build by the help of Machine Learning techniques. Through the data we have it make us easy to find the patterns among the relation of occurrence of crime. With the data we have, we undergo with preprocessing like removing null values and delete unwanted data. We got accuracy 86%. We use bar graphs, pie charts etc… for easily understand about the concept. This research and work would help to society effectively.

VIII.FUTURE SCOPE

Future research directions could focus on enhancing the predictive capabilities of crime prediction models by incorporating additional data sources, such as social media activity, weather patterns, and urban infrastructure. Moreover, exploring advanced machine learning techniques, such as deep learning and ensemble methods, may further improve the accuracy and robustness of predictive models in this domain.

REFERENCES

1) Yadav, S., Timbadia, M., Yadav, A., Vishwakarma, R., & Yadav, N. (2017, April). Crime pattern detection, analysis & prediction. In Electronics, Communication and AerospaceTechnology (ICECA), 2017 International conference of (Vol. 1, pp. 225- 230). IEEE.

2) Shamsuddin, N. H. M., Ali, N. A., & Alwee, R. (2017, May). An overview on crime prediction methods. In Student Project Conference (ICT-ISPC), 2017 6th ICT International (pp. 1-5). IEEE.

3) Sivaranjani, S., Sivakumari, S., & Aasha, M. (2016, October). Crime prediction and forecasting in Tamilnadu using clustering approaches. In Emerging Technological Trends (ICETT), International Conference on (pp. 1-6). IEEE

4) Sathyadevan, S., & Gangadharan, S. (2014, August). Crime analysis and prediction using data mining. In Networks & Soft Computing (ICNSC), 2014 First International Conference on (pp. 406-412). IEEE.

5) Nath, S. V. (2006, December). Crime pattern detection using data mining. In Web intelligence and intelligent agent technology workshops, 2006. wi-iat 2006 workshops. 2006 ieee/wic/acm international conference on (pp. 41-44). IEEE

6) Chen, Hsinchun, Wingyan Chung, Jennifer Jie Xu, Gang Wang, Yi Qin, and Michael Chau.

7) "Crime data mining: a general framework and some examples." computer 37, no. 4 (2004): 50-56.