COMPARISON OF MACHINE LEARNING ALGORITHMS FOR DETECTING CRIME HOTSPOTS

Minor project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

T. SWATHI PRATHYUSHA	(19UECS0983)	**(VTU13701)**
Y. SWETHA	(19UECS1072)	**(VTU15945)**
T. SREYAAW	(19UECS0931)	**(VTU13780)**

Under the guidance of
V.JAGANRAJA, B.Tech.,M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A Grade
CHENNAI 600 062, TAMILNADU, INDIA

June , 2022

CERTIFICATE

It is certified that work contained in the project report titled ”COMPARISON OF MACHINE LEARNING ALGORITHMS FOR DETECTING CRIME HOTSPOTS” by T. SWATHI PRATHYUSHA (19UECS0983), Y. SWETHA (19UECS1072), T. SREYAAW (19UECS0931) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
V.JaganRaja
Assisant professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science & Technology
June,2022

Signature of Head of the Department
Dr. V. Srinivasa Rao
Professor & Head
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science & Technology
June,2022

DECLARATION

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

(Signature)
T. SWATHI PRATHYUSHA
Date: / /

(Signature)
Y. SWETHA
Date: / /

(Signature)
T. SREYAAW
Date: / /

APPROVAL SHEET

This project report entitled ”COMPARISON OF MACHINE LEARNING ALGORITHMS FOR DETECTING CRIME HOTSPOTS” by T.SWATHI PRATHYUSHA (19UECS0983), Y.SWETHA (19UECS1072), T.SREYAAW (19UECS0931) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

V.JaganRaja, B.Tech.,M.E.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Dean & Head, Department of Computer Science & Engineering Dr.V.SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **V. JAGANRAJA,B.TECH., M.E.,**for his cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr.V. ASHOK KUMAR, M.Tech., Ms.C. SHYAMALA KUMARI, M.E., Ms.S.FLORENCE, M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

T. SWATHI PRATHYUSHA	**(19UECS0983)**
Y. SWETHA	**(19UECS1072)**
T. SREYAAW	**(19UECS0931)**

ABSTRACT

Crime is one of the biggest social problem in our society. Every day, a great number of crimes are perpetrated in the same or different areas. One of the major concerns that is always intensifying and becoming more complex is crime. Because criminal patterns and behaviour's are always changing, it's tough to describe them. In recent years, it has been clear that artificial intelligence has grown increasingly significant in virtually all businesses, including crime prediction. It is necessary to have a proper database of all crimes committed, as this knowledge will be beneficial in the future. From a strategic standpoint, the ability to forecast future crimes based on location, time, and pattern, among other factors, can be a significant source of information. Predicting future crimes accurately is a difficult challenge in today's environment, since the number of crimes continues to rise. As a result, crime prediction tools are critical for identifying and preventing future crimes. Predicting the crime in before helps in preventing the crime from occurring and reduces the crime rate in society. Data is collected from sources and analyzed and compared theoretically and practically. Some popular methods in machine learning like Decision Tree classifier , Random forest, Naive Bayes are applied in the collected data to determine their effectiveness in analyzing and preventing crime. They are evaluated based on the accuracy and confusion matrix

Keywords: Crime rate, Artificial intelligence, Decision tree, Naïve Bayes, Random forest, Accuracy, Confusion matrix

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

CNN	Convolution Neural Network
DTC	Decision Trees Classifier
NBC	Naive Bayes Classifier
RFC	Random Forest Classifier
SVM	Support Vector Machine

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

One of the most conspicuous and worrying aspects of our society is crime. Every day, a large number of violent crimes are attempted, and these everyday crimes have made life difficult for ordinary people. As a result, deterring these crimes is an important duty.The systematic effort to identifying crime patterns and trends is known as crime prediction.There are several sorts of crime, including theft, burglary, and robbery, as well as crime of aggression (homicides, assaults, and rapes).The ability to foresee future crimes can aid law enforcement agencies in preventing them before they happen. From a strategic standpoint, the capacity to predict any crime based on time and location aids in delivering important information to law enforcement authorities.

Crime is inherently unpredictable; it is influenced by a variety of circumstances such as poverty and employment.It is not random or uniform.Without any computational support, crime analysts find it incredibly difficult to perceive such massive amounts of data .However, because crime is expanding at an alarming rate, precisely anticipating crime is a difficult undertaking. Methods for detecting and reducing future crimes, such as crime prediction and analysis, are critical. As a result, nave Bayes is used in conjunction with machine learning models such as decision trees and Random Forest. The main aim is to show the feasibility and benefits of machine learning algorithms in identifying violent crimes in a certain area.

1.2 Aim of the project

To study the dataset and preprocess the data by removing duplicate and null values and remove the redundant attributes using feature selection method.

1.3 Project Domain

MACHINE LEARNING

Machine learning is a type of artificial intelligence that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

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

The Naive Bayes classification algorithm is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions. The independence assumptions often do not have an impact on reality. Therefore they are considered as naive.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable.

1.4 Scope of the Project

To study the data set and pre-process the data by removing duplicate and null values and remove the redundant attributes using feature selection method. After pre-processing the data set apply Decision tree ,Random Forest ,Naive Bayes,algorithm for predicting the crime type that occurs at given place and time Evaluate the performance of these algorithms in predicting the crime type and compare the accuracies of these algorithms.

Chapter 2

LITERATURE REVIEW

[1] Wajiha Safat Et Al.” Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques,” IEEE Access, IEEE, 2021.
Different machine learning techniques were used in this study, including logistic regression, SVM, K-Nearest Neighbors, Decision Tree, extreme Gradient Boosting, and time series. The model shows crime predicting and future crimes. This method of implementation is complex and causes an overall distortion.

[2] Umair Muneer butt Et Al.” SpatioTemporal Crime Hotspot Detection and Prediction: A Systematic Literature Review”, in IEEE, IEEE access, 2020.
The main purpose of this research is to gather, synthesise, and evaluate the state of the art for spatio-temporal crime hotspot detection and prediction approaches by conducting a complete literature review . It is simple to implement but doesn’t give real time results.

[3]Nicolas Esquivel Et Al.” Spatio - Temporal Prediction of Baltimore Crime events using Neural networks”, in IEEE, IEEE access, 2020.
A Convolutional Neural Network are suggested in this paper to predict the existence of crime events in Baltimore (USA).This project is Built-in Error Handling but a lot of pre-development education is needed to implement this project.

[4] Xinge han Et Al.” Risk Prediction of Theft Crimes in Urban communities: An integrated model of Long Short Term Memory”, in IEEE, IEEE access, 2020.
A Convolutional Neural Network (CNN) and a Long-Short Term Memory network are suggested in this paper to predict the existence of crime events in Baltimore It delivers high quality results but less secure and requires High Embedding extraction time.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The Existing system is to forecast crime rates, all that is needed is previous crime data. The data from the study is being used to evaluate the prediction skills of various machine learning algorithms. Incorporating variables improves the model's accuracy. As a result, criminology theories should be employed to forecast future criminal activity. Traditional methods identify crime hot spot locations by looking at the distribution of crime cases in the past and assuming that the pattern would repeat itself in the future. This assumption is inadequate for predicting long-term stable crime rates and crimes in different places. It is unable to predict future crime rates with any certainty. It's tough to pinpoint precise crime patterns.

3.1.1 Disadvantages

•Although this model gave better accuracy in the current scenario where the surge in cases is continuous, in a few cases like where the daily cases are fluctuating randomly, this model cannot predict with the above-mentioned accuracy.
•This model cannot predict Future Crime so it is inefficient for future prediction process.

3.2 Proposed System

There are two sets of data: one for training and one for testing. As machine learning models, we employ Decision Tree, Random Forest, and Naive Bayes. These models are trained and tested using the dataset. The dataset includes criminal features that can be used to analyse different sorts of crimes based on their location, time, year, and date. Finally, based on location and time, we'll establish the types of crimes that are occuring. The accuracy rate is used to evaluate the prediction model's performance. The suitable methodology is used to generate and test all of the methods. The accuracy that may be reached with this method is greater than 64and testing phases. The method is also far less complicated than the current system.

3.2.1 Advantages

•The approach can quickly spot commonalities between instances and also Improves classification prediction accuracy. Also, it is easy to implement.
•It is cost efficient and easy to implement.
•It helps the government to warn the officials of that particular area to get alert and take necessary precautions to avoid the crimes.

3.3 Feasibility Study

3.3.1 Economic Feasibility

Economic feasibility refers to how our idea relates to the economy and its demands, as well as how effective it is in meeting those demands. Whether the project can meet the project's basic needs and has sufficient value for further processing. To work on our project in a cost-effective and convenient manner.Techniques, software, and hardware requirements should be economically beneficial and cost effective in order to achieve completion and improved results. Data sets are clearly evaluated to find out the crime hot spots in country.
These are cost-effective and can be obtained with little effort.Our endeavour necessitates a fundamental data sets that the information related to the crimes happening in the country location wise and with the time, date,district. After collecting the data the data sets are pre-processed. thus, they are cost-effective and can be used in other ways

3.3.2 Technical Feasibility

Economic feasibility refers to how our idea relates to the economy and its demands, as well as how effective it is in meeting those demands. Whether the project can meet the project's basic needs and has sufficient value for further processing. To work on our project in a cost-effective and convenient manner.Techniques, software, and hardware requirements should be economically beneficial and cost effective in order to achieve completion and improved results. crime hot spots are examined and

the results are presented using simple machine learning algorithms.
These are cost-effective and can be obtained with little effort.Collecting is simple and inexpensive; thus, they are cost-effective and can be used in other ways is available and understandable with the usage needed. Python's required tools and libraries are included in the most recent version of Jupyter. This Jupyter software is simple to use.

3.3.3 Social Feasibility

The project is socially feasible and effective, and there are no issues that need to be addressed.Crimes happening can have a negative impact on social and physical interactions with society, leading to protectorate in certain situations. Our project concept revolves upon low-cost, high quality outputs.This helps in a variety of ways to alleviate some of the inconveniences that many individuals face. This project concept could be a small, new social change with a large number of beneficiaries.Furthermore, a socially feasible and practical method of minimising the crimes that are happening around us. crimes having increase the fear in the people and reduces the interactive of the people and reduce the socialization among people.

3.4 System Specification

3.4.1 Hardware Specification

• 4 GB RAM

• Better GPU(For performance)

3.4.2 Software Specification

• Anaconda 2021.11

• System:64 bit OS, x64 processor

3.4.3 Standards and Policies

Anaconda Prompt

Anaconda prompt is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.The anaconda prompt has many number of IDE's which make the coding easier. The UI can also be implemented in python.

Standard Used: ISO/IEC 27001

Jupyter

It's like an open source web application that allows us to share and create the documents which contains the live code, equations, visualizations and narrative text. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

Standard Used: ISO/IEC 27001

MATPLOTLIB

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

Chapter 4

METHODOLOGY

4.1 General Architecture

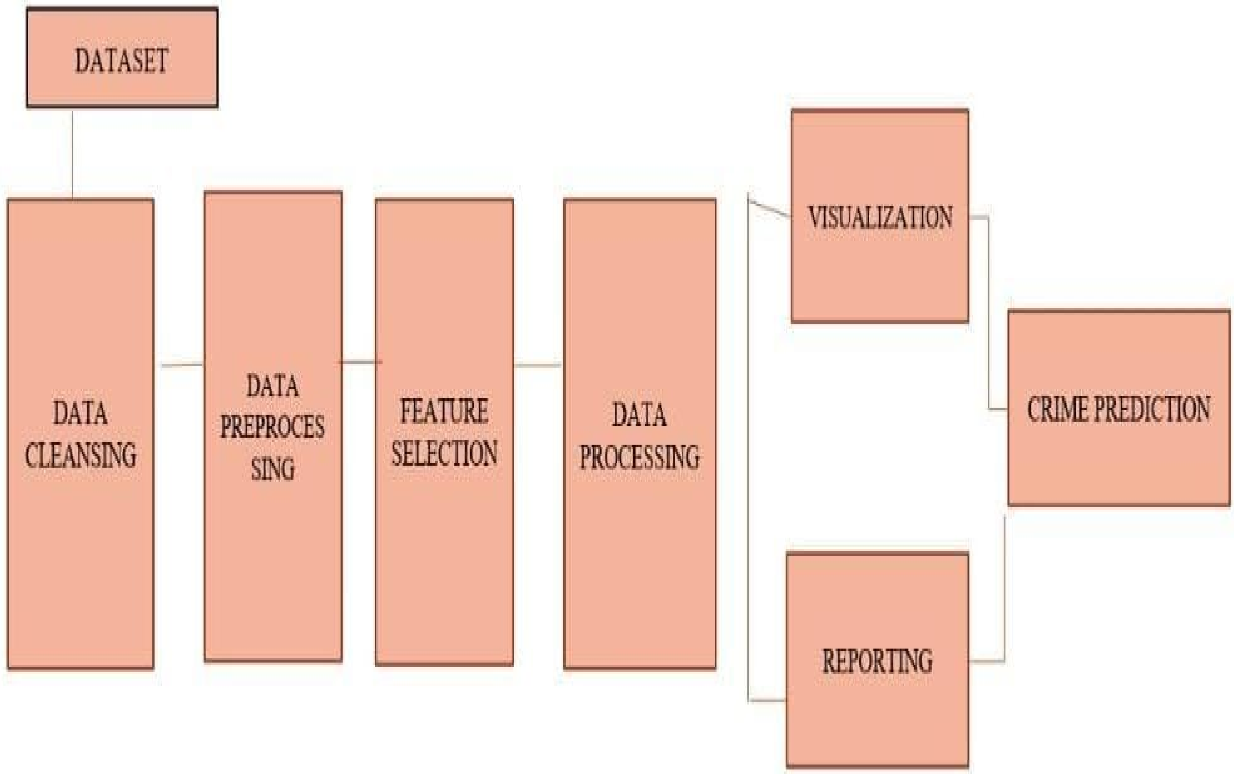


Figure 4.1: **Architecture diagram of detecting crime hotspots**

In the above fig 4.1 the general architecture of our project has being shown it has various phases like data cleansing, data pre-processing,feature extraction,data visualization,prediction and reporting.

4.2 Design Phase

4.2.1 Data Flow Diagram

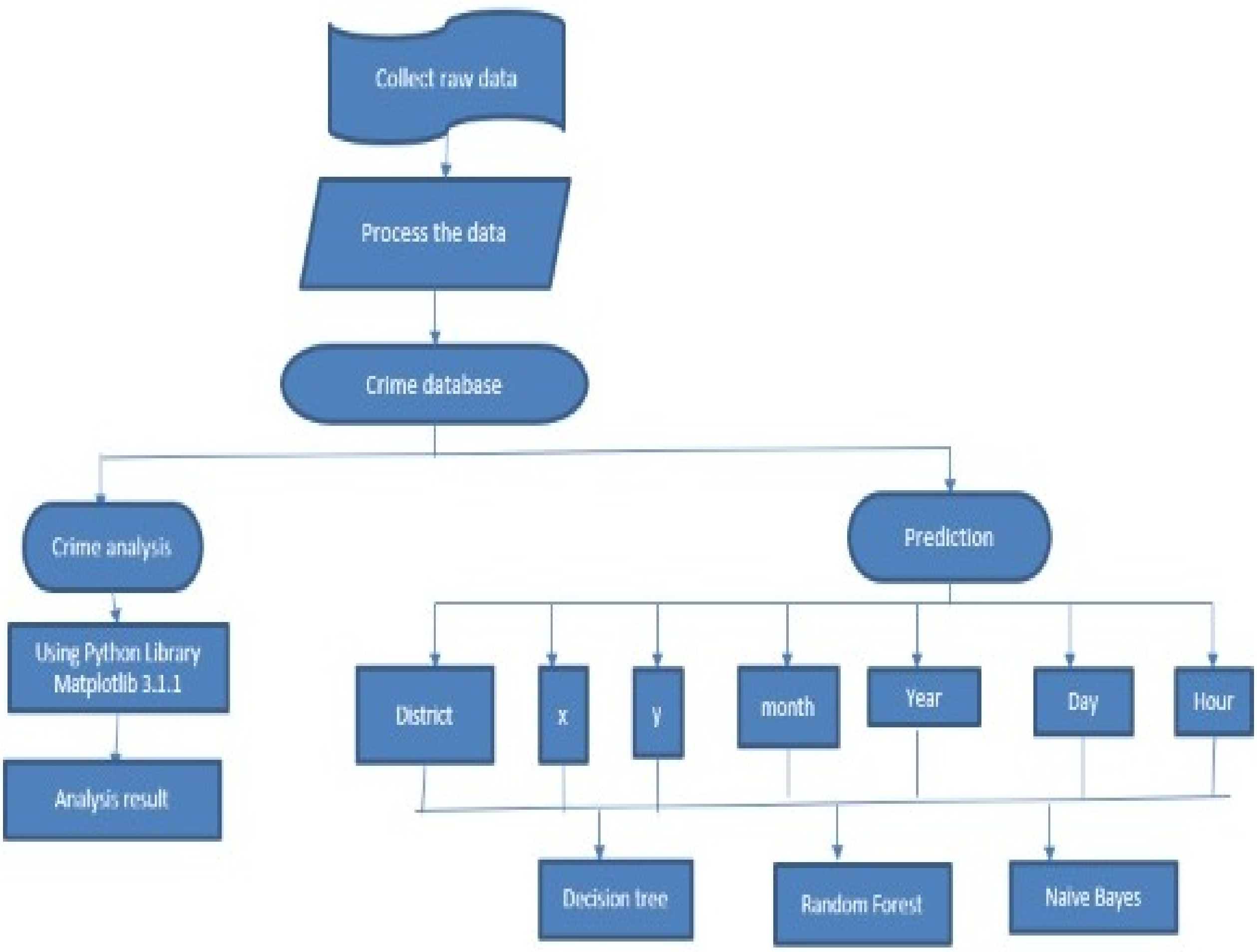


Figure 4.2: **Data flow Diagram of detecting crime hotspots**

This fig 4.2 describes the data flow diagram of our project. It shows the clear process how the project has been executed, this starts from collecting of data and ends with the deciding about the best machine learning algorithm.

4.2.2 Use Case Diagram

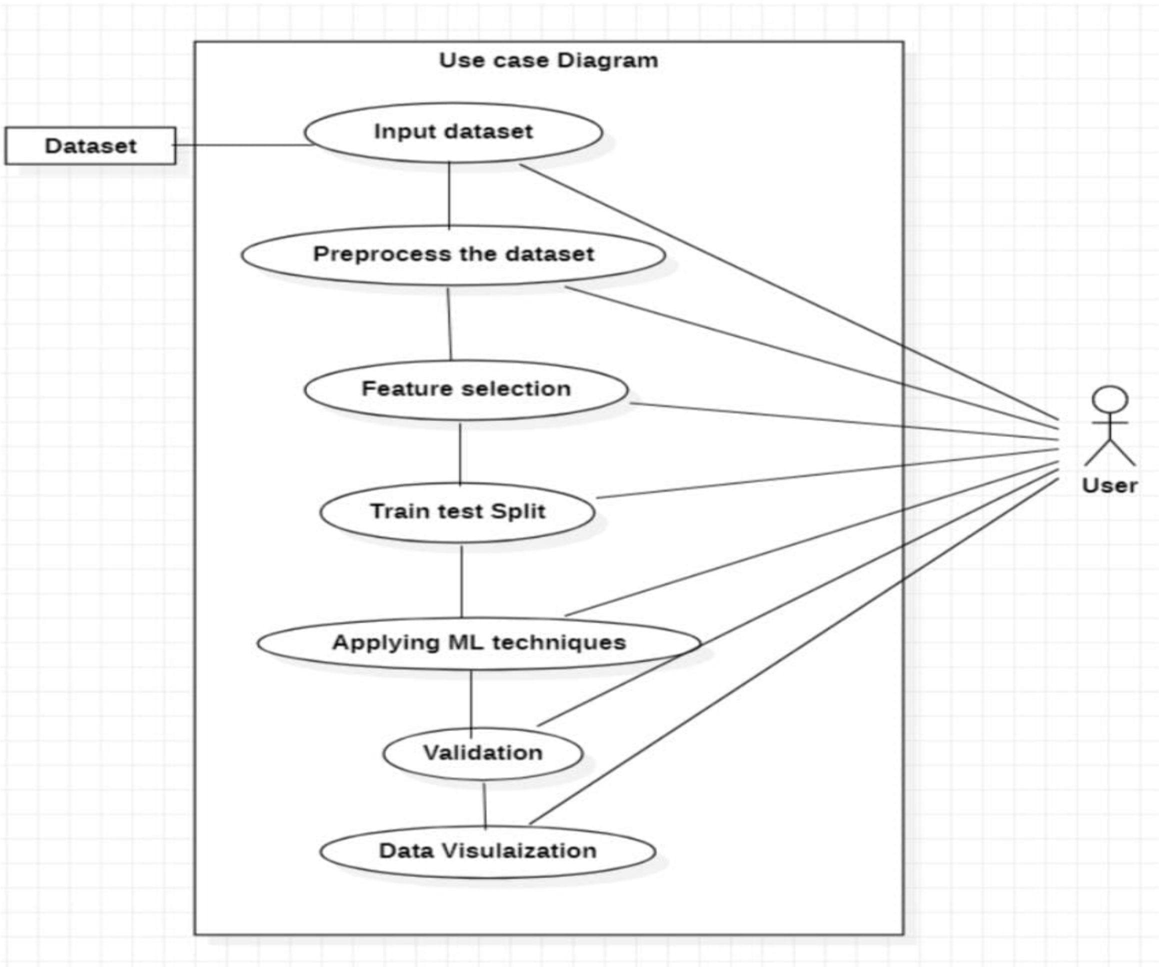


Figure 4.3: **Use Case Diagram of detecting crime hotspots**

In this fig 4.3 the use case diagram of our project is shown,Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors.

4.2.3 Class Diagram

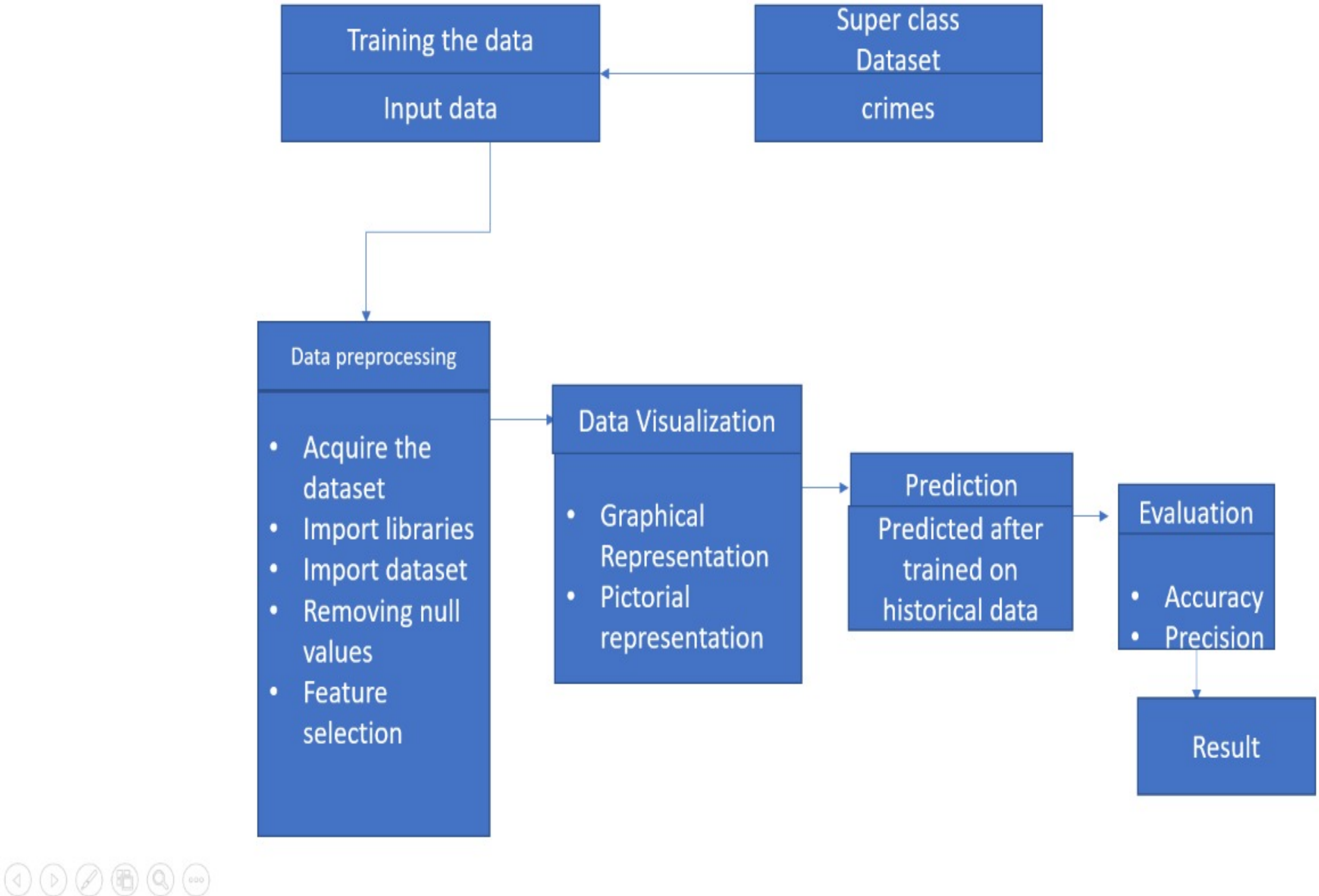


Figure 4.4: **Class Diagram of detecting crime hotspots**

In this fig 4.4 the class diagram of our project has been showed, Class diagram describes the attributes and operations of a class and also the constraints imposed on the system.an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language.

4.2.4 Sequence Diagram

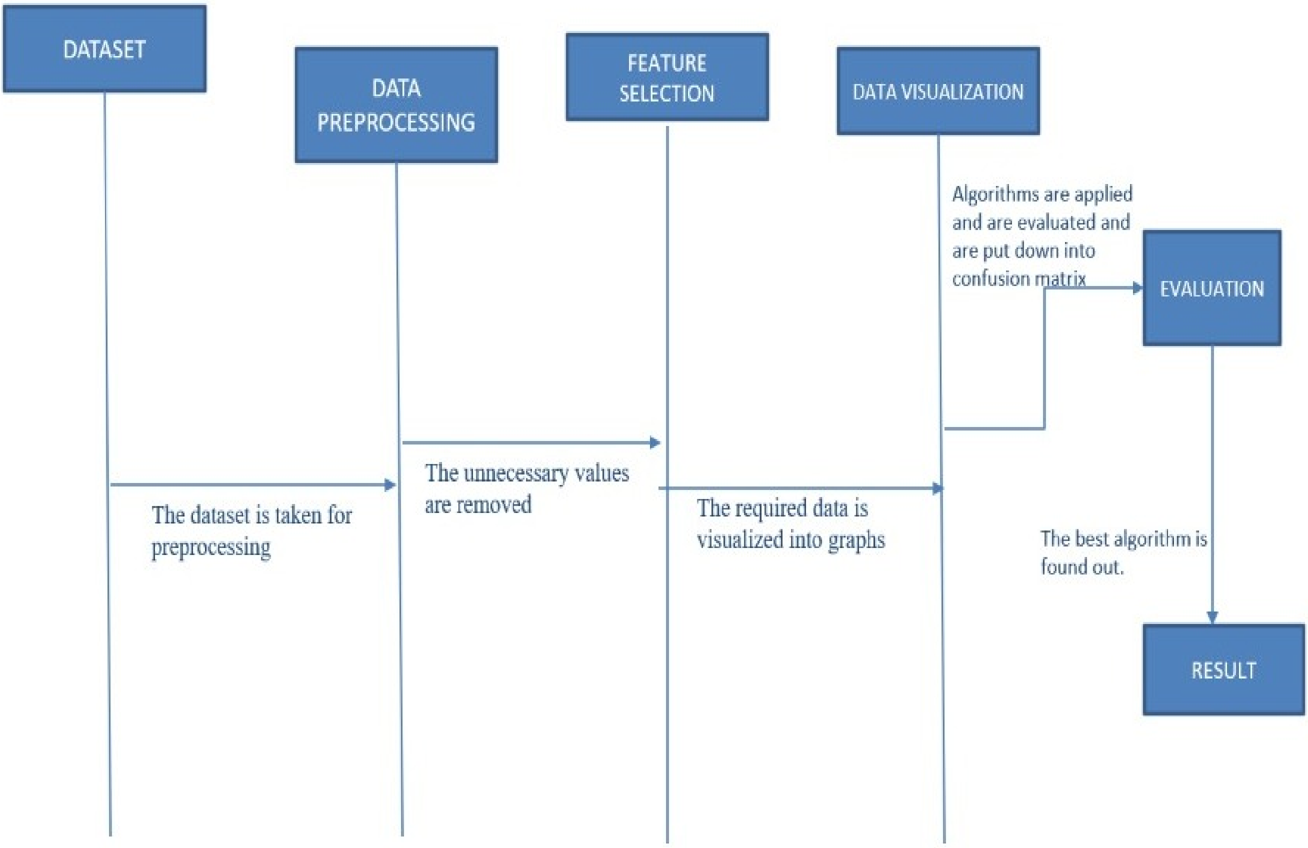


Figure 4.5: **Sequence Diagram of detecting crime hotspots**

In in the fig 4.5 the sequence diagram is indicated here, A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. The reason the sequence diagram is so useful is because it shows the interaction logic between the objects in the system in the time order that the interactions take place.

4.2.5 Collaboration diagram

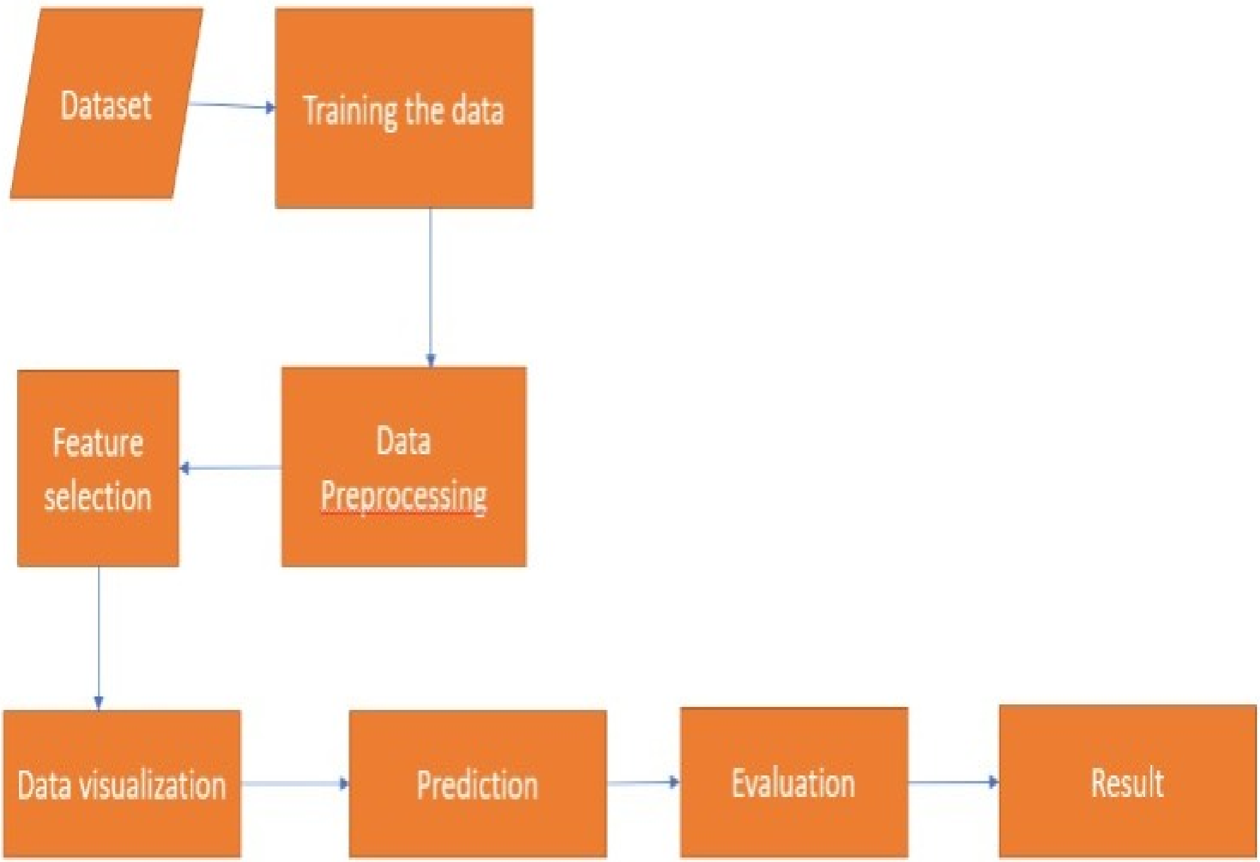


Figure 4.6: **Collaboration Diagram of detecting crime hotspots**

In the fig 4.6 the collaboration diagram of our project has been projected here. collaboration diagram has flow different phases like dataset, training,preprocessing, data visualization, feature selection.A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions.

4.2.6 Activity Diagram

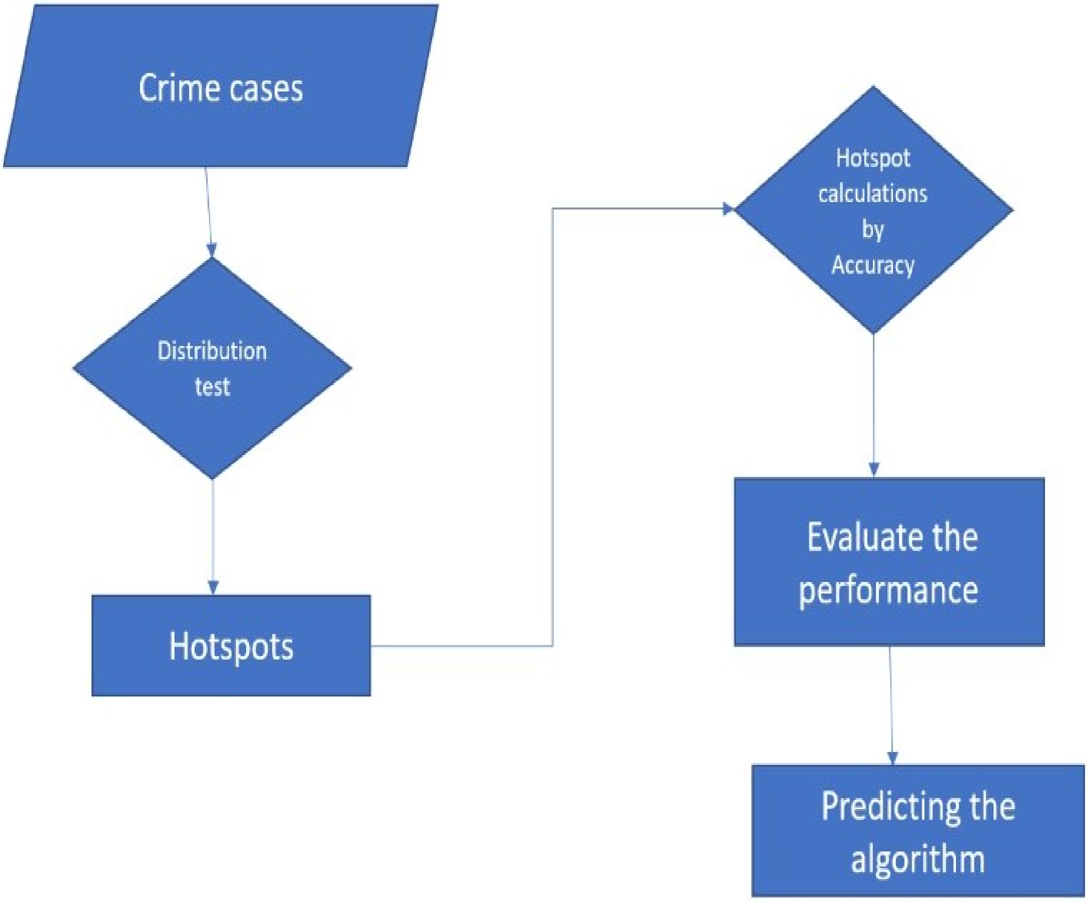


Figure 4.7: **Activity Diagram of detecting crime hotspots**

In fig 4.7 activity diagram of our project has been showed. An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

Step1: Open Google colab and select to the new notebook.
Step2: Connect the google drive to Google Colab to import the data set.
Step3: Install the required libraries needed for the program.

Step4: Import required libraries like keras,pandas,matplot lib,Imagedatagenerator.
Step5: Unzip the dataset from the drive using zip.extractall().
Step6: Preprocess the data and create the functions as create imagemetadata.csv ,getImageLabels,getImage,and Build Model.
Step7: Preprocess the data and create the train and valid data csv files.
Step8: Construct the model with the help of ResNet50 and train the model.
Step9: Plot the training and validation accuracy values , loss values.
Step10: Plot the confusion mtrix and also confusion matrix.
Step11: After the traing and validation train the classifier with the different deep learning algorithms like sgd,knn,decision tree and logistic regression and calculate the accuracy of the each classifier.
Step12:Based on the accuracy, model we will test the input image with high accuracy value.
Step13: Model will predict the output as fractured or Not fractured as output.

4.3.2 Pseudo Code

• Connect google drive the Google colab.to the drive.
• Import all the libraries needed for the code.
• Set the path of the Zip file and Unzip it.
• (i)Create a function as [create-images-metadata-csv] and set the image data as dictonary data type and study labels as positive nad negative.
(ii)Set image data[catageory] as path ,count and label.
(iii)The csv file for data and validation.
• (i)create the function as getImageLabels and set labels as empty.
(ii) using for loop set tqdm(dataframe.iterrows()) as i, data (iii) set labels as np.asarray(labels) and return labels.
• (i)Create the function getImage and declare variable as data frame and size. (ii)Set Images as empty list.
(iii)Using for loop set tqdm(dataframe.iterrows()) as i,data and using try block read the csv file.
(iv) Set np.mean(Images) as mean and np.std(Images) as std and (Images - mean) / std as Images and return Images.
• (i)For data preprocessing set img-width, img-height as 224 and image as train-data-dir as well as for valid-data-dir.

(ii)Call the ImageDataGenerator and test-datagen.
(iii)create the images-metadata.csv files for train and valid data.
• (i)read the valid-image-data.csv and train-image-data.csv and set dd as empty dictonary.
(ii)set train-image-df as dd[train] and do same for valid also.
(iii)call the getImageLabels and getImage,,traindatagen.flow,train-datagen.fit.
• (i)construct the model and set len(train-images) as nb-train-samples and do same for validation.
(ii)Call the build model, model.compile and model.fit.Save the model.
(iii)call the model.evaluate ans set to ev.
• (i)Plot the ”Training Accuracy vs Validation Accuracy”
(ii)Plot the ”Training Accuracy vs Validation Loss”
(iii)Plot the ”Training Accuracy vs Validation MSE”
• (i)Import confusion matrix, classification report, cohen kappa score from sklearn. metrics.
(ii)Print the confusion matrix.
(iii)Plot the ROC curve with False positive and True positive.
• (i)Now we need to verify some machine learning algorithms.
(ii) set ridge as classifier and call the train-model and also call confusion matrix.
(iii)Repeat the above step for RFC(Random Forest Classifier),Decision Tree,Navie Bayes.
• (i)Display the activation map by call the function display-class-activation-map.

4.4 Module Description

4.4.1 DATA PRE-PROCESSING

Data pre-processing is the process of preparing raw data so that it can be used in a machine learning model. It is the first and most important phase in the development of an AI and machine learning model. When working on a machine learning project, we don’t always reveal all of the relevant information. Keeping in mind that any activity involving information must be cleaned and organised. As a result, the data pre-processing task is employed for this. Genuine data, for the most part, comprises noises, missing values, and may be in an unsuitable format that cannot be used directly for machine learning models. Data preprocessing is a necessary step in cleaning data and making it usable for a certain purpose.

To make machine learning model, the primary thing required is a data set as a machine learning model totally chips away at information. The gathered information for a specific issue in a legitimate arrangement is known as the data set. Here, in this case, the data set is taken from the Kaggle website. As this data set contains duplicate values and null values and the data is obtained without any errors. The zero values in the data set indicate that no crime cases were recorded on that particular day. In this work, a python code is used for pre-processing, by which the category data will be obtained as output. This facilitates the training of models. Rather than this everything was fine with the data that has been taken from the web.

4.4.2 FEATURE SELECTION

The number of features in the data set is increased by developing new features from existing ones and then deleting the old ones. And the reduction should be done so that the new reduced set of features can summarise the majority of the information in the original collection of features. It is carried out in order to refine the data by removing irrelevant information from the data set. It also reduces the number of dimensions. The data has been cleansed after these procedures.

After that, the necessary data for training the models is gathered. Dates, category, descript, dayofweek, district, resolution, and address are among the attributes in the data collection.. The format of date we taken is YY-MM-DD HH:MM:SS and instead of date attribute, it is included extra attributes such as month, year, day, hour to our dataset and corresponding values were added in it by extracting month, year, day, hour values from date. This is done for better processing. After extraction, date attribute is no longer needed. So, it is dropped fromdataset.Category attribute has string values. These values are converted to some numerical values based on category type. Next unwanted attributes from data set are removed. These are descript, dayofweek, resolution, address attributes. After removing we left with remaining columnscategory, district, month, year, day, hour. Theseare used for training purpose.

4.4.3 EVALUATION METHOD

Here we use accuracy for evaluating the machine learning algorithms that were used in this project. The best algorithm will be chosen among all algorithms based on accuracy for crime rates prediction in this model.
accuracy:One parameter for evaluating classification models is accuracy. Informally, accuracy refers to the percentage of correct predictions made by our model.
The following is the formal definition of accuracy:

Accuracy = Number of correct predictions / Total number of predictions

The following formula can be used to calculate accuracy in terms of positives and negatives for binary classification:

Accuracy = (TP+TN) / (TP+TN+FP+FN)

4.5 Steps to execute/run/implement the project

4.5.1 Collect the data set

1. Collect the data set that is required for executing the project.
2. This data set is collected from the website called Kaggale.
3. The collected data set is stored in the computer we are using for processing.

4.5.2 Upload data in drive

1. After collecting the data set then the data set is uploaded into the google Drive.
2. Then the file appears in google drive so that is used to access the data so quickly for the execution of the project.

4.5.3 Run the code

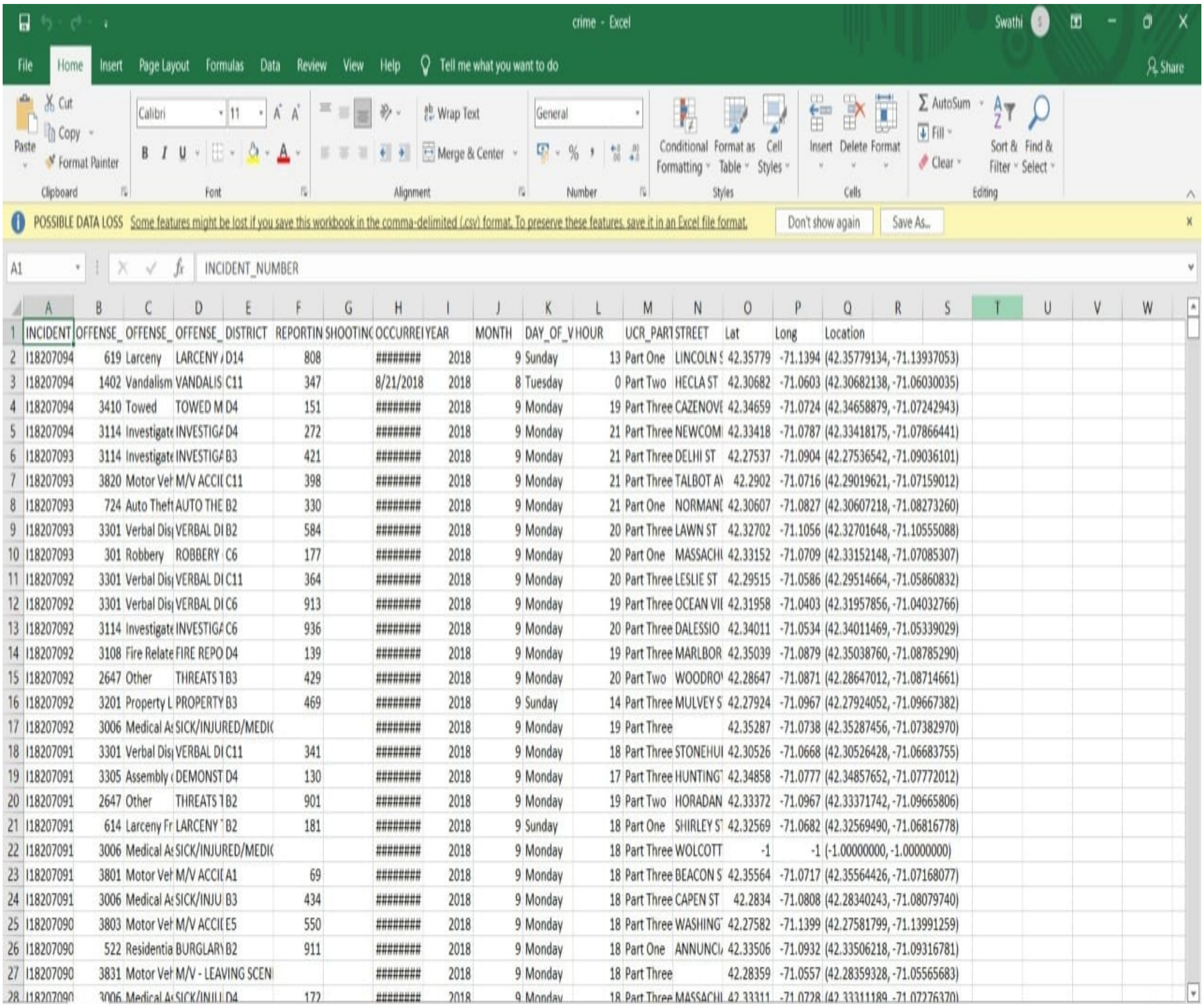
1. After the code is uploaded in google drive then the code is run the software jupyter.
2. Jupyter gets the information about the data set from the google drive and executes the code and produces the required output for the program.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	INCIDENT	OFFENSE_	OFFENSE_	OFFENSE_	DISTRICT	REPORTIN	SHOOTIN(	OCCURREI	YEAR	MONTH	DAY_OF_V	HOUR	UCR_PART	STREET	Lat	Long	Location
2	I18207094	619	Larceny	LARCENY /	D14	808		########	2018	9	Sunday	13	Part One	LINCOLN S	42.35779	-71.1394	(42.35779134, -71.13937053)
3	I18207094	1402	Vandalism	VANDALIS	C11	347		8/21/2018	2018	8	Tuesday	0	Part Two	HECLA ST	42.30682	-71.0603	(42.30682138, -71.06030035)
4	I18207094	3410	Towed	TOWED M	D4	151		########	2018	9	Monday	19	Part Three	CAZENOVE	42.34659	-71.0724	(42.34658879, -71.07242943)
5	I18207094	3114	Investigate	INVESTIGA	D4	272		########	2018	9	Monday	21	Part Three	NEWCOM	42.33418	-71.0787	(42.33418175, -71.07866441)
6	I18207093	3114	Investigate	INVESTIGA	B3	421		########	2018	9	Monday	21	Part Three	DELHI ST	42.27537	-71.0904	(42.27536542, -71.09036101)
7	I18207093	3820	Motor Veh	M/V ACCI(	C11	398		########	2018	9	Monday	21	Part Three	TALBOT AV	42.2902	-71.0716	(42.29019621, -71.07159012)
8	I18207093	724	Auto Theft	AUTO THE	B2	330		########	2018	9	Monday	21	Part One	NORMAND	42.30607	-71.0827	(42.30607218, -71.08273260)
9	I18207093	3301	Verbal Dis	VERBAL DI	B2	584		########	2018	9	Monday	20	Part Three	LAWN ST	42.32702	-71.1056	(42.32701648, -71.10555088)
10	I18207093	301	Robbery	ROBBERY	C6	177		########	2018	9	Monday	20	Part One	MASSACHU	42.33152	-71.0709	(42.33152148, -71.07085307)
11	I18207092	3301	Verbal Dis	VERBAL DI	C11	364		########	2018	9	Monday	20	Part Three	LESLIE ST	42.29515	-71.0586	(42.29514664, -71.05860832)
12	I18207092	3301	Verbal Dis	VERBAL DI	C6	913		########	2018	9	Monday	19	Part Three	OCEAN VIE	42.31958	-71.0403	(42.31957856, -71.04032766)
13	I18207092	3114	Investigate	INVESTIGA	C6	936		########	2018	9	Monday	20	Part Three	DALESSIO	42.34011	-71.0534	(42.34011469, -71.05339029)
14	I18207092	3108	Fire Relate	FIRE REPO	D4	139		########	2018	9	Monday	19	Part Three	MARLBOR	42.35039	-71.0879	(42.35038760, -71.08785290)
15	I18207092	2647	Other	THREATS T	B3	429		########	2018	9	Monday	20	Part Two	WOODRO'	42.28647	-71.0871	(42.28647012, -71.08714661)
16	I18207092	3201	Property L	PROPERTY	B3	469		########	2018	9	Sunday	14	Part Three	MULVEY S	42.27924	-71.0967	(42.27924052, -71.09667382)
17	I18207092	3006	Medical As	SICK/INJURED/MEDI(				########	2018	9	Monday	19	Part Three		42.35287	-71.0738	(42.35287456, -71.07382970)
18	I18207091	3301	Verbal Dis	VERBAL DI	C11	341		########	2018	9	Monday	18	Part Three	STONEHUI	42.30526	-71.0668	(42.30526428, -71.06683755)
19	I18207091	3305	Assembly (	DEMONST	D4	130		########	2018	9	Monday	17	Part Three	HUNTING'	42.34858	-71.0777	(42.34857652, -71.07772012)
20	I18207091	2647	Other	THREATS T	B2	901		########	2018	9	Monday	19	Part Two	HORADAN	42.33372	-71.0967	(42.33371742, -71.09665806)
21	I18207091	614	Larceny Fr	LARCENY T	B2	181		########	2018	9	Sunday	18	Part One	SHIRLEY ST	42.32569	-71.0682	(42.32569490, -71.06816778)
22	I18207091	3006	Medical As	SICK/INJURED/MEDI(				########	2018	9	Monday	18	Part Three	WOLCOTT	-1	-1	(-1.00000000, -1.00000000)
23	I18207091	3801	Motor Veh	M/V ACCI(	A1	69		########	2018	9	Monday	18	Part Three	BEACON S	42.35564	-71.0717	(42.35564426, -71.07168077)
24	I18207091	3006	Medical As	SICK/INJU	B3	434		########	2018	9	Monday	18	Part Three	CAPEN ST	42.2834	-71.0808	(42.28340243, -71.08079740)
25	I18207090	3803	Motor Veh	M/V ACCI(	E5	550		########	2018	9	Monday	18	Part Three	WASHING'	42.27582	-71.1399	(42.27581799, -71.13991259)
26	I18207090	522	Residentia	BURGLARY	B2	911		########	2018	9	Monday	18	Part One	ANNUNCI/	42.33506	-71.0932	(42.33506218, -71.09316781)
27	I18207090	3831	Motor Veh	M/V - LEAVING SCEN				########	2018	9	Monday	18	Part Three		42.28359	-71.0557	(42.28359328, -71.05565683)
28	I18207090	3006	Medical As	SICK/INJU	D4	172		########	2018	9	Monday	18	Part Three	MASSACHU	42.33311	-71.0728	(42.33311189, -71.07276370)

Figure 5.1: **Input image of detecting crime hotspots**

5.1.2 Output Design

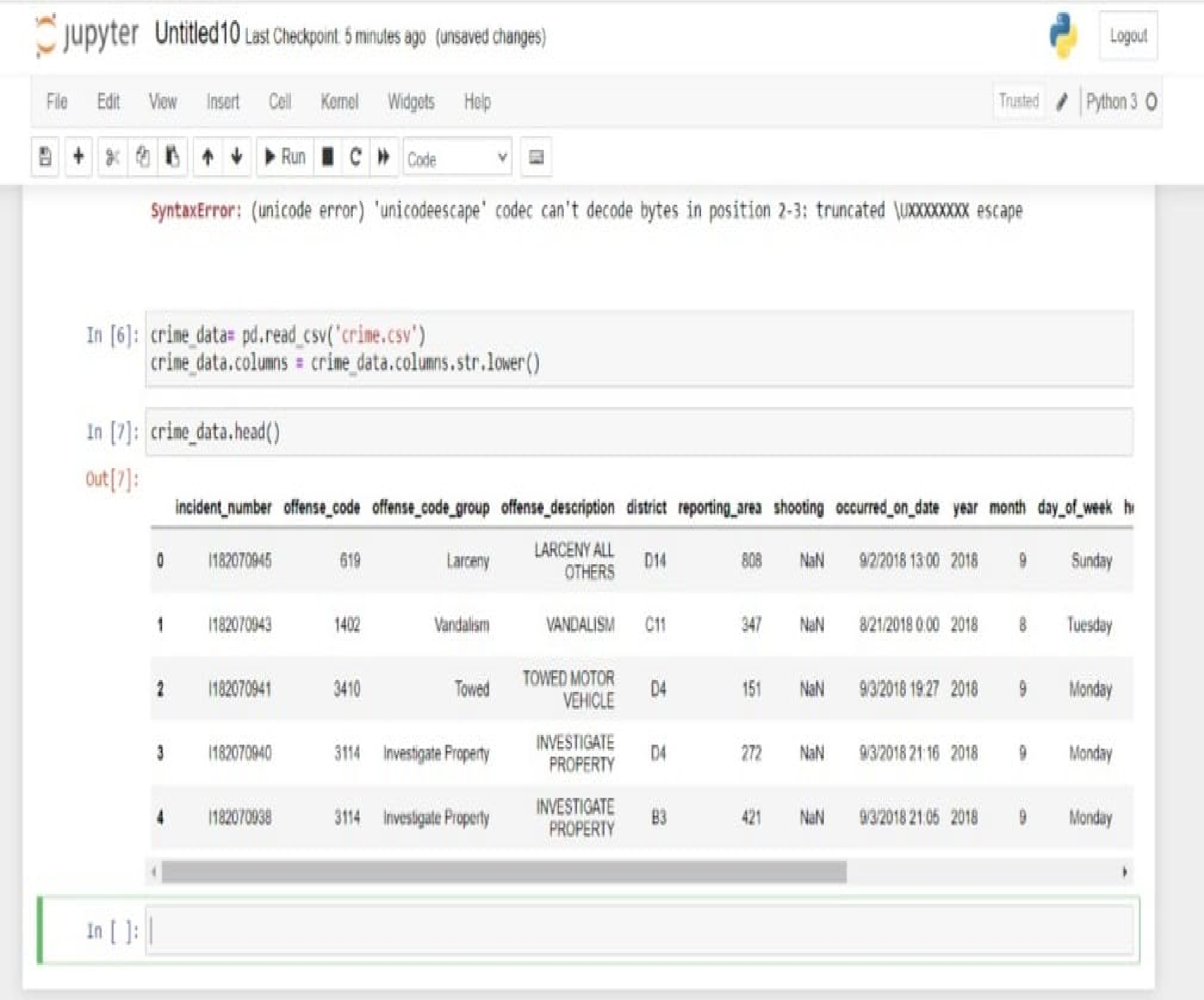


Figure 5.2: **Output of detecting crime hotspots**

5.2 Testing

```
%%html
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
import pylab
from IPython.core.display import HTML
from sklearn.preprocessing import LabelEncoder
from wordcloud import WordCloud
from joblib import dump
from sklearn.preprocessing import MinMaxScaler
from sklearn.naive_bayes import GaussianNB
df = pd.read_csv("Dataset/train.csv.zip")
df.head()
df.info()
df.shape
HTML("<h1 style='color:green;text-decoration:underline;text-align:center;margin:35pt;'>No of Rows
Available in Training Dataset is</h1>".format(df.shape[0]))
duplicated_rows = df.duplicated().sum()
HTML("<h1 style='color:#954e4e;text-decoration:underline;text-align:center;margin:35pt;'>No of Duplicated
Rows Available in Training Dataset is </h1>".format(duplicated_rows))
df.drop_duplicates(inplace=True)
duplicated_rows = df.duplicated().sum()
HTML("<h1 style='color:#954e4e;text-decoration:underline;text-align:center;margin:35pt;'>No of Duplicated
Rows Available in Training Dataset is</h1>".format(duplicated_rows))
null_values = df.isnull().sum()
```

Figure 5.3: **Testing code of detecting crime hotspots**

Figure 5.4: **Testing output of detecting crime hotspots**

5.3 Types of Testing

5.3.1 Unit testing

```
crime_data= pd.read_csv('crime.csv')
crime_data.columns = crime_data.columns.str.lower()
crime_data.head()
```

Out[7]:

	incident_number	offense_code	offense_code_group	offense_description	district	reporting_area	shooting	occurred_on_date	year	month	day_of_week
0	I182070945	619	Larceny	LARCENY ALL OTHERS	D14	808	NaN	9/2/2018 13:00	2018	9	Sunday
1	I182070943	1402	Vandalism	VANDALISM	C11	347	NaN	8/21/2018 0:00	2018	8	Tuesday
2	I182070941	3410	Towed	TOWED MOTOR VEHICLE	D4	151	NaN	9/3/2018 19:27	2018	9	Monday
3	I182070940	3114	Investigate Property	INVESTIGATE PROPERTY	D4	272	NaN	9/3/2018 21:16	2018	9	Monday
4	I182070938	3114	Investigate Property	INVESTIGATE PROPERTY	B3	421	NaN	9/3/2018 21:05	2018	9	Monday

Figure 5.5: **Output of unit testing for detecting crime hotspots**

5.3.2 Integration testing

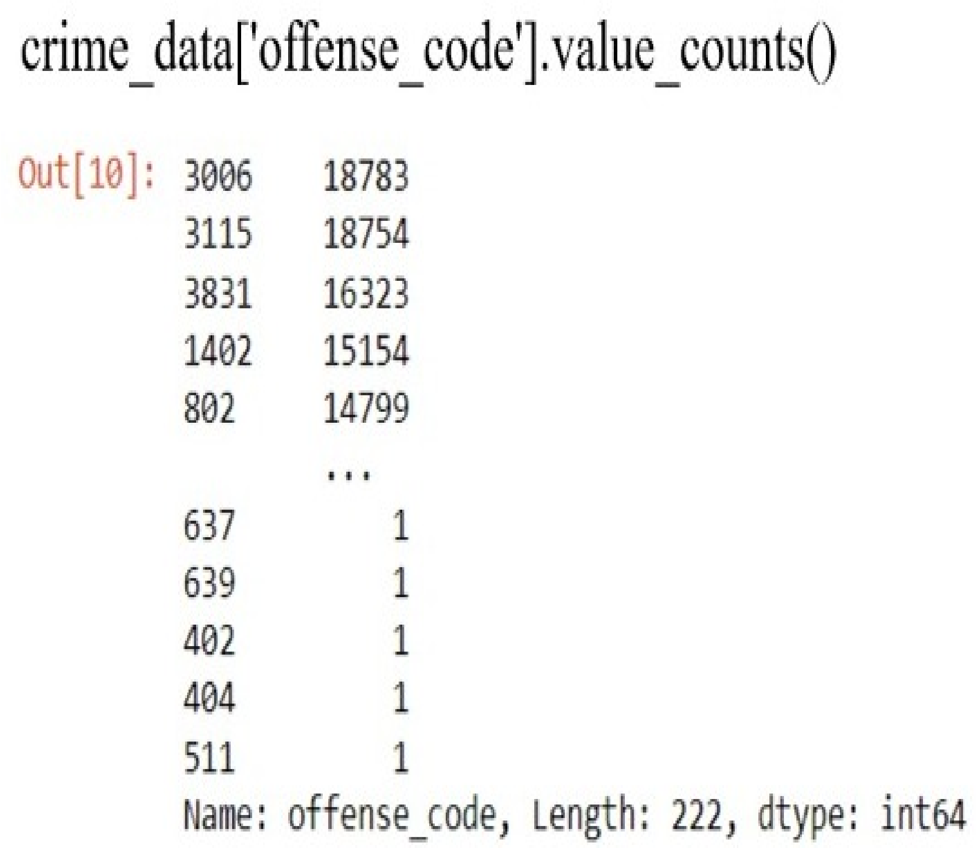


Figure 5.6: **Output of integration testing for detecting crime hotspots**

5.3.3 System testing

Input

```
\begin{lstlisting}
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
import pylab
from IPython.core.display import HTML
```

```python
from sklearn.preprocessing import LabelEncoder
from wordcloud import WordCloud
from joblib import dump
from sklearn.preprocessing import MinMaxScaler
from sklearn.naive_bayes import GaussianNB
df.head()
df.info()
df.shap
sns.set_style("Dark Grid")
```

5.3.4 Test Result

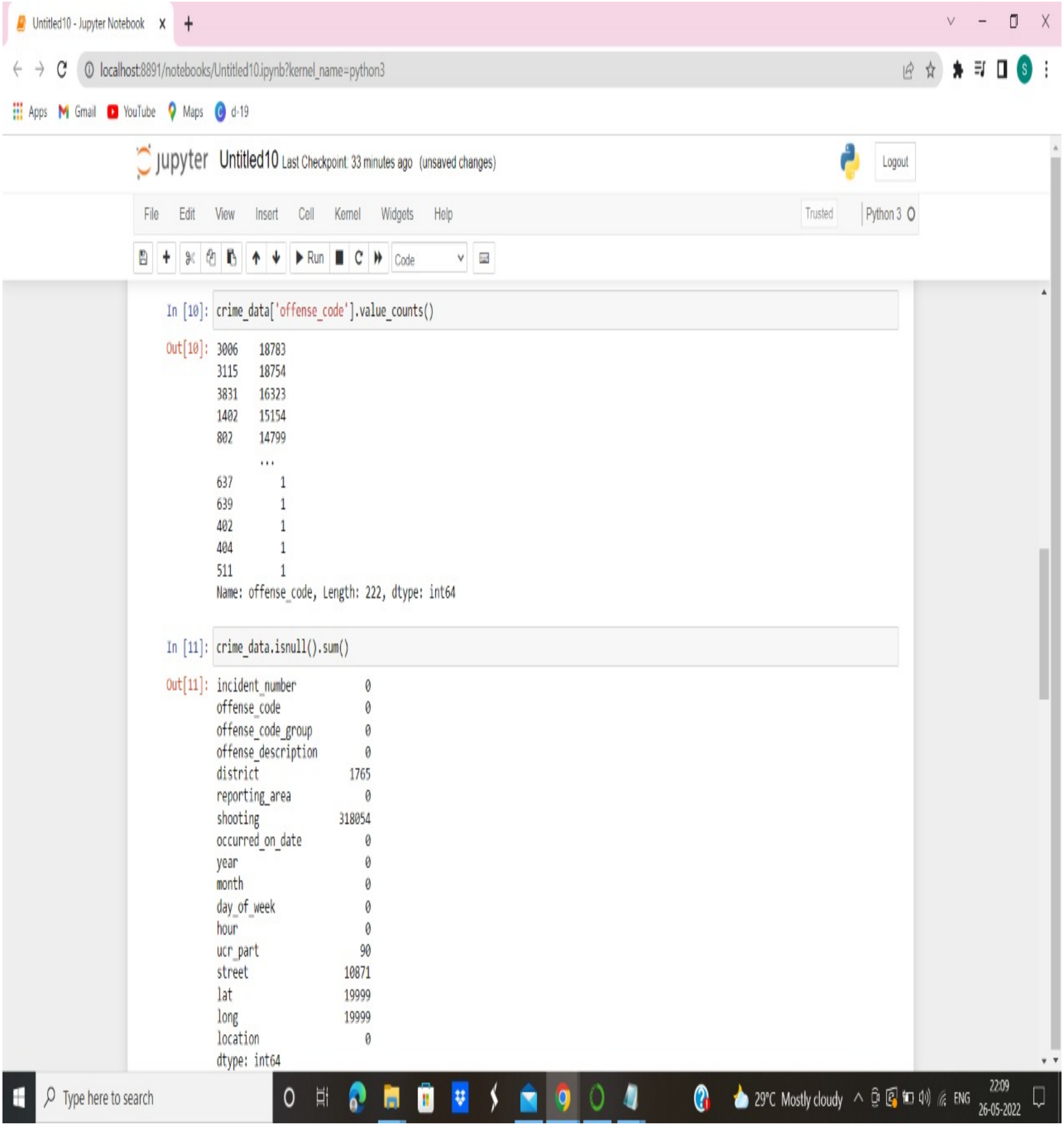


Figure 5.7: **Test Image of detecting crime hotspots**

5.3.5 Output

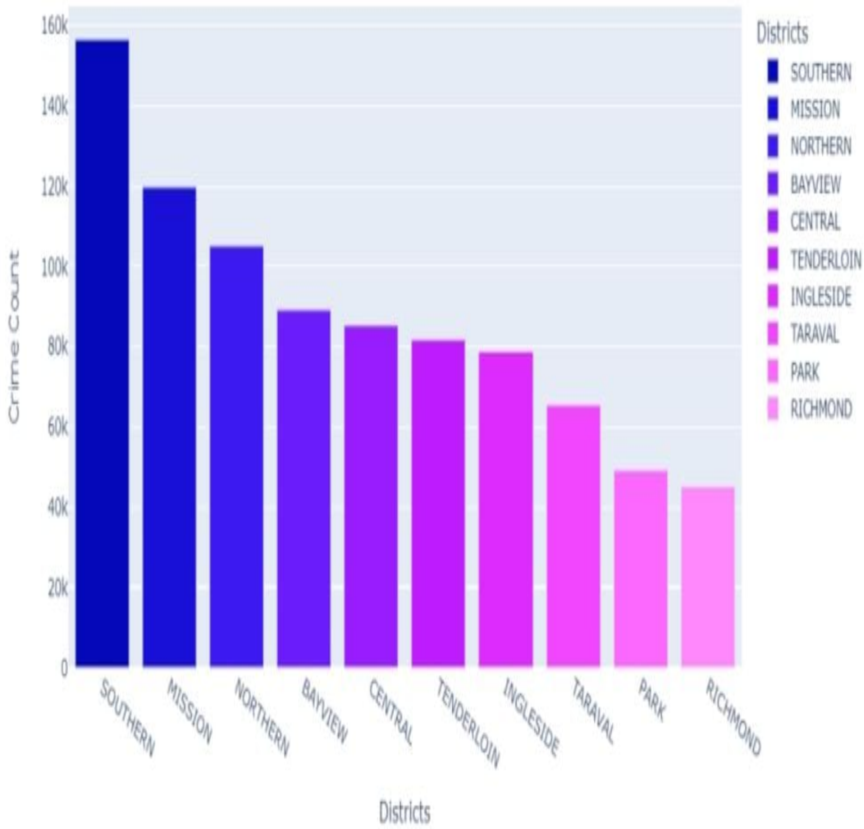


Figure 5.8: **Bar Graph of detecting crime hotspots**

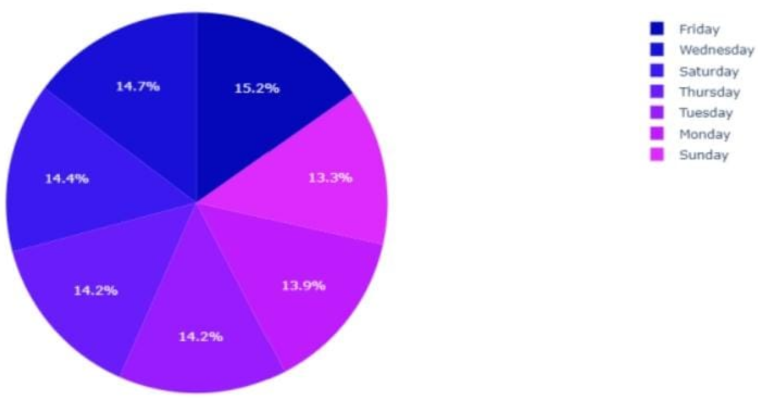


Figure 5.9: **Pie Chart of detecting crime hotspots**

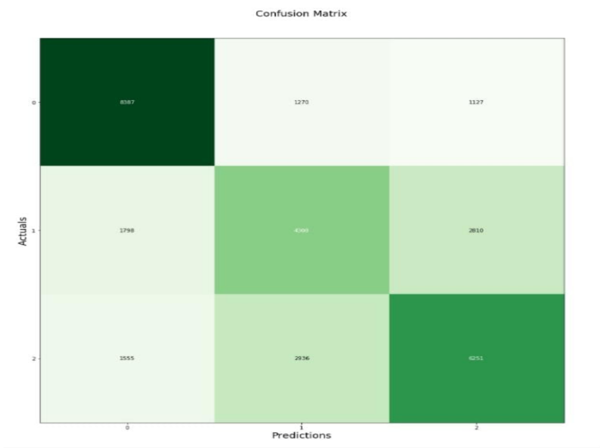


Figure 5.10: **Confusion Matrix for Decision Tree for detecting crime hotspots**

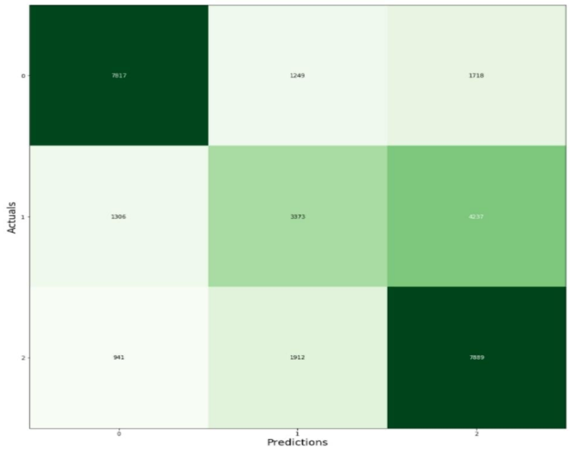


Figure 5.11: **Confusion Matrix for Random Forest Classifier for detecting crime hotspots**

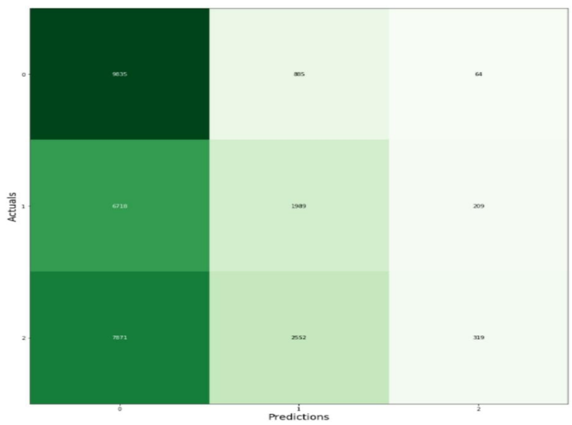


Figure 5.12: **Confusion Matrix for Navie Bayes Algorithm for detecting crime hotspots**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

There are two sets of data: one for training and one for testing. As machine learning models, we employ Decision Tree, Random Forest, and Naive Bayes. These models are trained and tested using the dataset. The dataset includes criminal features that can be used to analyse different sorts of crimes based on their location, time, year, and date. Finally, based on location and time, we'll establish the types of crimes that are occuring. The accuracy rate is used to evaluate the prediction model's performance. The suitable methodology is used to generate and test all of the methods. The accuracy that may be reached with this method is greater than 64and testing phases. The method is also far less complicated than the current system.

6.2 Comparison of Existing and Proposed System

Existing system

In the Existing system, To forecast crime rates, all that is needed is previous crime data. The data from the study is being used to evaluate the prediction skills of various machine learning algorithms. Incorporating variables improves the model's accuracy. As a result, criminology theories should be employed to forecast future criminal activity. Traditional methods identify crime hot spot locations by looking at the distribution of crime cases in the past and assuming that the pattern would repeat itself in the future. This assumption is inadequate for predicting long-term stable crime rates and crimes in different places. It is unable to predict future crime rates with any certainty. It's tough to pinpoint precise crime patterns. proposed system.

Proposed system

There are two sets of data: one for training and one for testing. As machine learning models, we employ Decision Tree, Random Forest, and Naive Bayes. These models are trained and tested using the data set. The data set includes criminal features that can be used to analyse different sorts of crimes based on their location, time, year, and date. Finally, based on location and time, we'll establish the types of crimes that are occurring. The accuracy rate is used to evaluate the prediction model's performance. The suitable methodology is used to generate and test all of the methods. The accuracy that may be reached with this method is greater than 64and testing phases. The method is also far less complicated than the current system.

6.3 Sample Code

```
%%html
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
from IPython.core.display import HTML
from joblib import load
df = pd.read_csv("Dataset/test_label_encoding.csv", header=None)
df.head()
X_test = df[[0,1,2]]
y_test = df[[3]]
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import classification_report,confusion_matrix, accuracy_score
dtree_model = load('dtree_model.joblib')
y_pred = dtree_model.predict(X_test)
accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
fig, ax = plot_confusion_matrix(conf_mat=conf_matrix, figsize=(16, 16), cmap=plt.cm.Greens)
plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Confusion Matrix', fontsize=18)
plt.show()
```

```
rf_40_model = load('rf_40_model.joblib')
y_pred = rf_40_model.predict(X_test)
accuracy_score(y_test, y_pred)
rf_60_model = load('rf_60_model.joblib')
y_pred = rf_60_model.predict(X_test)
conf_matrix = confusion_matrix(y_true=y_test, y_pred=y_pred)
fig, ax = plot_confusion_matrix(conf_mat=conf_matrix, figsize=(16, 16), cmap=plt.cm.Greens)
plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Confusion Matrix', fontsize=18)
plt.show()
accuracy_score(y_test, y_pred)
nb_model = load('nb_model.joblib')
y_pred = nb_model.predict(X_test)
accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_true=y_test, y_pred=y_pred)
fig, ax = plot_confusion_matrix(conf_mat=conf_matrix, figsize=(16, 16), cmap=plt.cm.Greens)
plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Gaussian Naive Bayes - Confusion Matrix', fontsize=18)
plt.show()
```

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

Crimes pose a severe threat to human society, safety, and long-term development, and must thus be addressed. Authorities in charge of investigations frequently request computational forecasts and predictive systems that i mprove crime analytics in order to improve city safety and security and aid in crime prevention. All the three methods are successfully implemented. The type of crime and in which area the crime is occurring most has been predicted successfully. Here we achieved an improved predictive accuracy for crimes by implementing different machine learning algorithms on crime dataset. Among the different algorithms, Decision Tree achieves the maximum accuracy on dataset.

• We got 63 percent Accuracy for Decision Tree Algorithm.

• We got 62 perecent Accuracy for Random Forest Algorithm.

• We got 40 percent Accuracy for Naïve Bayes Algorithm.

• We discovered that Decision Tree Algorithm is the best of the three models that were employed in the analysis. As a result, we may infer that Decision Tree Algorithm is very effective in Predicting Crime rate.

7.2 Future Enhancements

Crime is the dominant factor of our society. Predicting crime is difficult task. These machine learning algorithms help in predicting the crime in an efficient way. Future crime prediction can be done through Arima model. In future it can be applied to many schemes.

Chapter 8

PLAGIARISM REPORT

Report Title:	Plagarism Report
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/79618215291ecfda4186fa4564a2a9a0342ca1654326350
Report Generated Date:	04 June, 2022
Total Words:	550
Total Characters:	3951
Keywords/Total Words Ratio:	0%
Excluded URL:	No
Unique:	92%
Matched:	8%

Figure 8.1: **Plagarism Report of detecting crime hotspots**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
%%html
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
import pylab
from IPython.core.display import HTML
from sklearn.preprocessing import LabelEncoder
from wordcloud import WordCloud
from joblib import dump
from sklearn.preprocessing import MinMaxScaler
from sklearn.naive_bayes import GaussianNB
df = pd.read_csv("Dataset/train.csv.zip")
df.head()
df.info()
df.shape
HTML("<h1 style='color:green;text-decoration:underline;text-align:center;margin:35pt;'>No of Rows
Available in Training Dataset is {0}</h1>".format(df.shape[0]))
duplicated_rows = df.duplicated().sum()
HTML("<h1 style='color:#954e4e;text-decoration:underline;text-align:center;margin:35pt;'>No of
    Duplicated
Rows Available in Training Dataset is {0}</h1>".format(duplicated_rows))
df.drop_duplicates(inplace=True)
duplicated_rows = df.duplicated().sum()
HTML("<h1 style='color:#954e4e;text-decoration:underline;text-align:center;margin:35pt;'>No of
    Duplicated
Rows Available in Training Dataset is {0}</h1>".format(duplicated_rows))
null_values = df.isnull().sum()
```

```
display_table = '''<h1 style='color:#954e4e;text-decoration:underline;text-align:center;margin:35pt;
    '>No of
Null Values Available in Training Dataset is </h1>
 <table style='font-family:Roboto,sans-serif;font-size:1.8rem;'>
 <thead>
 <tr>
 <th>Column Name</th>
 <th>Null Values Count</th>
 </tr>
 </thead>
 <tbody>
 '''
for k,v in null_values.items():
 display_table = display_table+ '''
 <tr>
 <td>{0}</td>
 <td>{1}</td>
 </tr>
 '''.format(k, v)
display_table = display_table + '''
 </tbody>
 </table>
'''
HTML(display_table)
category_counts = df["Category"].value_counts()
category_counts
fig = px.bar(category_counts, x=category_counts.index, y=category_counts.values, title="Major Crimes
    in
Sanfrancisco",
 color=category_counts.index, color_discrete_sequence=px.colors.sequential.Agsunset, height=1000,
 labels= {
 'index': "Crime Category",
 'y': 'Crime Count'
 }
 )
fig.show()
district_wise_crime_counts = df["PdDistrict"].value_counts()
district_wise_crime_counts
fig = px.bar(district_wise_crime_counts, x=district_wise_crime_counts.index,
y=district_wise_crime_counts.values,
 title="Top Dangerous District wise Crime Count",
 color=district_wise_crime_counts.index, color_discrete_sequence=px.colors.sequential.Plotly3,
 labels = {
 'index': "Districts",
 'y' : "Crime Count"
 })
fig.show()
region_wise_crime_counts = df["Address"].value_counts().head(20)
region_wise_crime_counts
fig = px.pie(region_wise_crime_counts, names=region_wise_crime_counts.index,
```

```
values=region_wise_crime_counts.values,
title="Top 20 Region wise Crime Count",
 color=region_wise_crime_counts.index, color_discrete_sequence=px.colors.sequential.Aggrnyl,
 labels = {
 'index': "Regions",
 'y' : "Crime Count"
 }
 )
fig.show()
day_wise_crime_counts = df["DayOfWeek"].value_counts().head(20)
fig = px.pie(day_wise_crime_counts, names=day_wise_crime_counts.index,
values=day_wise_crime_counts.values,
 title="Day wise Crime Count",
 color=day_wise_crime_counts.index, color_discrete_sequence=px.colors.sequential.Plotly3,
 labels = {
 'index': "Days",
 'y' : "Crime Count"
 }
 )
fig.show()
date = pd.to_datetime(df["Dates"])
date
df["Month"] = date.dt.month
df["Year"] = date.dt.year
df["Day"] = date.dt.day
df["Hour"] = date.dt.hour
df.head()
df.drop("Dates", axis=1, inplace=True)
pylab.rcParams['figure.figsize'] = (14.0, 8.0)
larceny = df[df['Category'] == "LARCENY/THEFT"]
assault = df[df['Category'] == "ASSAULT"]
drug = df[df['Category'] == "DRUG/NARCOTIC"]
vehicle = df[df['Category'] == "VEHICLE THEFT"]
vandalism = df[df['Category'] == "VANDALISM"]
burglary = df[df['Category'] == "BURGLARY"]
with plt.style.context('fivethirtyeight'):
 ax1 = plt.subplot2grid((3,3), (0,0), colspan=3)
 ax1.plot(df.groupby('Hour').size(), 'ro-')
 ax1.set_title ('All crimes')
 start, end = ax1.get_xlim()
 ax1.xaxis.set_ticks(np.arange(start, end, 1))
 ax2 = plt.subplot2grid((3,3), (1, 0))
 ax2.plot(larceny.groupby('Hour').size(), 'o-')
 ax2.set_title ('Larceny/Theft')

 ax3 = plt.subplot2grid((3,3), (1, 1))
 ax3.plot(assault.groupby('Hour').size(), 'o-')
 ax3.set_title ('Assault')

 ax4 = plt.subplot2grid((3,3), (1, 2))
```

```
 ax4.plot(drug.groupby('Hour').size(), 'o-')
 ax4.set_title ('Drug/Narcotic')

 ax5 = plt.subplot2grid((3,3), (2, 0))
 ax5.plot(vehicle.groupby('Hour').size(), 'o-')
 ax5.set_title ('Vehicle')

 ax6 = plt.subplot2grid((3,3), (2, 1))
 ax6.plot(vandalism.groupby('Hour').size(), 'o-')
 ax6.set_title ('Vandalism')

 ax7 = plt.subplot2grid((3,3), (2, 2))
 ax7.plot(burglary.groupby('Hour').size(), 'o-')
 ax7.set_title ('Burglary')

 pylab.gcf().text(0.5, 1.03,
 'San Franciso Crime Occurence by Hour',
 horizontalalignment='center',
 verticalalignment='top',
 fontsize = 28)

plt.tight_layout(2)
plt.show()
plt.rcParams['figure.figsize'] = (15, 15)
plt.style.use('fast')
txt = "".join(t for t in df['Descript'].head(1000))
wc = WordCloud(width = 1500, height = 1500).generate(txt)
plt.title('Description of the Crime', fontsize = 20)
plt.imshow(wc)
plt.axis('off')
plt.show()
df.info()
px.box(df, x='Day', color_discrete_sequence=px.colors.sequential.RdBu)
px.box(df, x='Month', color_discrete_sequence=px.colors.sequential.Plasma)
px.box(df, x='Year', color_discrete_sequence=px.colors.sequential.Agsunset)
df.drop(['Resolution', 'Descript', 'Address','DayOfWeek'], inplace=True, axis=1)
df.head()
category_counts = df["Category"].value_counts()
category_counts
df1 = df.copy()
df = df1.copy()
df = df[
 (df['Category'] == 'DRUG/NARCOTIC')
 | (df['Category'] == 'VEHICLE THEFT')
 | (df['Category'] == 'VANDALISM') ]
le = LabelEncoder()
df["Category"] = le.fit_transform(df.Category)
df
le_pd = LabelEncoder()
df["PdDistrict"] = le_pd.fit_transform(df["PdDistrict"])
```

```
df.head()
mms = MinMaxScaler()
df[["PdDistrict", "X", "Y", "Month", "Day", "Year", "Hour"]] = mms.fit_transform(df[["PdDistrict", "X", "Y",
"Month", "Day", "Year", "Hour"]])
df.head()
X = df[["PdDistrict", "X", "Y"]]
y = df["Category"]
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,random_state=0, stratify=y)
type(X_test)
X_test
y_test
test = np.column_stack((X_test, y_test))
test
np.savetxt("Dataset/test_label_encoding.csv", test, delimiter=",")
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import classification_report,confusion_matrix, accuracy_score
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
dtree_model = dtree.fit(X_train,y_train)
dump(dtree_model, 'dtree_model.joblib')
from sklearn.ensemble import RandomForestClassifier
rf_40 = RandomForestClassifier(n_estimators=100,min_samples_split=100, max_features="log2")
rf_40_model = rf_40.fit(X_train,y_train)
dump(rf_40_model, 'rf_40_model.joblib')
rf_60 = RandomForestClassifier(n_estimators=60,min_samples_split=100 )
rf_60_model = rf_60.fit(X_train,y_train)
dump(rf_60_model, 'rf_60_model.joblib')
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb_model = nb.fit(X_train,y_train)
dump(nb_model, 'nb_model.joblib')
```

9.2 Poster Presentation

Figure 9.1: **Poster Presentation of detecting crime hotspots**

References

[1] L. Lin,W. J. Liu, andW.W. Liao, "Comparison of random forest algorithm and space- time kernel density mapping for crime hotspot prediction," Prog. Geogr., vol. 37, no. 6, pp. 761-771, 2018.

[2] B. Wang, P. Yin, A. L. Bertozzi, P. J. Brantingham, S. J. Osher, and J. Xin, "Deep learning for real-time crime forecasting and its ternarization," Chin. Ann. Math., B, vol. 40, no. 6, pp. 949–966, Nov. 2019.

[3] S. Chackravarthy, S. Schmitt, and L. Yang, "Intelligent crime anomaly detection in smart cities using deep learning," in Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC), Philadelphia, PA, USA, Oct. 2018, pp. 399–404.

[4] G. R. Nitta, B. Y. Rao, T. Sravani, N. Ramakrishiah, and M. BalaAnand, "LASSO- based feature selection and Naïve Bayes classifier for crime prediction and its type," Serv. Oriented Comput. Appl., vol. 13, no. 3, pp. 187-197, 2019.

[5] X. Hong, R. Lin, C. Yang, N. Zeng, C. Cai, J. Gou, and J. Yang, "Predicting Alzheimer's disease using LSTM," IEEE Access, vol. 7, pp. 80893-80901, 2019.

[6] L. Mou, P. Zhao, and Y. Chen, "Short-term traffic flow prediction: A long short-term memory model enhanced by temporal information," in Proc. 19th COTA Int. Conf. Transp. Prof. CICTP Transp. ChinaConnect.World, 2019, pp. 2411-2422.

[7] N. Ibrahim, S. Wang, and B. Zhao, "Spatiotemporal crime hotspots analysis and crime occurrence prediction," in International Conference on Advanced Data Mining and Applications. Springer, 2019, pp. 579– 588.

[8] G. R. Nitta, B. Y. Rao, T. Sravani, N. Ramakrishiah, and M. BalaAnand, "LASSO- based feature selection and Naïve Bayes classifier for crime predic-

tion and its type," Service Oriented Comput. Appl., vol. 13, no. 3, pp. 187–197, Sep. 2019.