

SURVEY ON CRIME ANALYSIS AND PREDICTION USING MACHINE LEARNING TECHNIQUES

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

PERALA TEJASWINI

(Reg. No. 38110402)

PENTAREDDY MANASVI

(Reg. No. 38110401)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

MARCH 2022



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act,1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, Chennai -600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Perala Tejaswini (Reg No. 38110402)**, **Pentareddy Manasvi (Reg No. 38110401)** who have done the project work as a team who carried out the project entitled “**SURVEY ON CRIME ANALYSIS AND PREDICTION USING MACHINE LEARNING**” under my supervision from nov 2021 to MAR 2022.

Internal Guide

MS.M.D. ANTO PRAVEENA., M.C.A., M.E., (Ph.D.,)

Head of the Department

Dr. L. LAKSHMANAN., M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I, **PENTAREDDY MANASVI (RegNo.38110401), PERALA TEJASWINI(Reg No. 38110402)** hereby declare that the professional Training Report on “**SURVEY ON CRIME ANALYSIS AND PREDICTION USING MACHINE LEARNING**” done by me under the guidance of **MS.M.D. ANTO PRAVEENA,,M.C.A.,M.E.,(Ph.D.,)** at Sathyabama Institute of Science and technology is submitted in partial fulfillment of the requirements of the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean of the Department, Department of Computer Science and Engineering** for providing us the necessary support and details at the right time during the progressive reviews.

We convey our thanks to **Dr. L. LAKSHMANAN., M.E., Ph.D., Head of the Department, Department of Computer Science and Engineering** for providing us the necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **MS.M.D. ANTO PRAVEENA.,M.C.A.,M.E.,(Ph.D).,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

ABSTARCT

Social media text analytics is the process of deriving information from text sources. Text analysis can be applied to any text-based dataset, including social media. Crime is a major problem faced today by society even we are finding in social media. Crimes have affected the quality of life and economic growth badly. It can identify the crime patterns and predict the crimes by detecting and analyzing the historical data. However, some crimes are unregistered and unsolved due to a lack of evidence. Thus, detecting crimes is a still challenging task. Some can use social media to detect crimes related activities. Because social media users sometimes convey messages related to his or her surrounding environment via social media message. It is proposed as a machine learning approach to detect the crimes and analyses it on its type. As the first step, we fetch the text messages using predefined keywords relating to the crimes. Then, after the preprocessing, we applied a support vector machine- based filtering approach to eliminate the noise. And then Random forest is used for classification . Then in the final stage, it analyses and categorize the crime type.

TABLE OF CONTENTS

Chapter. No	TITLE	Page. No
	ABSTRACT	v
	LIST OF FIGURES	viii
	ABBREVIATIONS	x
1	INTRODUCTION	1
1.1	OUTLINE OF THE PROJECT	1
1.2	OBJECTIVES OF PROJECT	4
1.3	DOMAIN INTRODUCTION	2
	1.3.1 BASICS OF PYTHON	2
	1.3.2 PYTHON FEATURES	2
2	AIM AND SCOPE	3
2.1	AIM OF PROJECT	3
2.2	SCOPE OF PROJECT	3
2.3	SOFTWARE REQUIREMENTS	3
2.4	HARDWARE REQUIREMENTS	3

3	METHODS AND ALGORITHMS USED	4
3.1	DATA COLLECTION	4
3.2	DATA CLEANING	4
3.3	DATA PREPROCESSING	4
3.4	FEATURE EXTRACTION	5
3.5	DATA PREPARATION	6
3.6	SECOND STAGE FILTERING	6
3.7	TESTING MODEL	7
3.8	PERFORMANCE EVALUATION	8
3.9	PREDICTION	9
	3.9.1 NLTK	10
	3.9.2 TENSORFLOW	10
	3.9.3 KERAS	10
3.10	SUPPORT VECTOR MACHINE	11
	3.10.1 WHAT IS SVM	12
	3.10.2 HOW DOES IT WORK	13
3.11	RANDOM FOREST ALGORITHM	18
	3.11.1 WORKING OF RFA	19
4	RESULTS	22
4.1	RESULTS	22
4.2	SCREEN SHORTS OF OUTPUT	23
5	CONCLUSION AND FUTURE WORK	24
5.1	CONCLUSION	24
5.2	FUTURE SCOPE	24
	REFERENCES	25

APPENDIX	26
A. SOURCE CODE	26

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	SYSTEM ARCHITECTURE	5
3.2	BLOCK DIAGRAM	7
3.3	WORK FLOW DIAGRAM	11
3.4	N-DIMENSIONAL SPACE	12
3.5	HYPER-PLANE A,B,C	13
3.6	HYPER-PLANE SEGREGATION	13
3.7	MAXIMIZING THE DISTANCE	13
3.8	IDENTIFYING THE HYPER-PLANE	14
3.9	CLASSIFYING THE CLASSES	14
3.10	ADDITIONAL FEATURES	15
3.11	ORIGINAL INPUT SPACE	16
3.12	BOOTSTRAPPING	19
4.1	SCREENSHOT OF OUTPUT	22
4.2	CRIME DETECTION REPORT	22
4.3	CRIME ANALYSIS REPORT	23

ABBREVIATIONS

ML	MACHINE LEARNING
SVM	SUPPORT VECTOR MACHINE
RFA	RANDOM FOREST ALGORITHM
NLTK	NATIONAL LANGUAGE TOOL KIT
NLP	NATIONAL LANGUAGE PROCESSING
ROI	RETURN ON INVESTMENT

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF THE PROJECT

Security is a very necessary aspect of life. Unless we are safe, our most important needs cannot be met. Security is therefore a requirement in human life that helps us to achieve our goals collectively or individually. Crimes are a social problem, which costs our society deeply in many aspects. The ability to identify unsafe areas for crime and identify the most recent crime in a particular location has become a growing concern for both local authorities and residents. On the other hand, people are always interested in improving safety and make reliable relationships with neighbors when living in a busy society. The prevalence of crime is one of the greatest challenges for societies around the world, particularly in metropolitan areas. There are more researches regarding social crimes in the world, but using social media, there are few types of research about crimes and their behavior. Therefore, the paper aims in presenting a prediction model (algorithm) by using the machine-learning technique, which is meant to possess a strong capability to predict crimes by factors of social media dataset using the Data Mining concept. Our main data source is social media. The main goal is to identify each hidden data source and predict results.

1.2 OBJECTIVES OF THE PROJECT

- The main objective of the project is to predict the crime rate and analyze the crime rate to be happened in future. Based on this Information the officials can take charge and try to reduce the crime rate.
- The concept of Multi Linear Regression is used for predicting the graph between the Types of Crimes (Independent Variable) and the Year (Dependent Variable)

- The system will look at how to convert crime information into a regression problem, so that it will help detectives in solving crimes faster.
- Crime analysis based on available information to extract crime patterns. Using various multi linear regression techniques, frequency of occurring crime can be predicted based on territorial distribution of existing data and Crime recognition.

1.3 DOMAIN INTRODUCTION

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

1.3.1 Basics of Python

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.

1.3.2 Python features

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language.

CHAPTER 2

AIM AND SCOPE

2.1 AIM OF THE PROJECT

- Crime detection is aimed for detecting the crime in social media platforms.
- It will be detecting the social media crime by using of machine learning concept. It used the data set for detecting the crime.

2.2 SCOPE OF THE PROJECT

- Future work, we planned to evaluate other machine learning algorithms, such as neural networks and decision trees. We also plan to implement the crime prediction approach using SVM and deep neural network approach.
- Also, we intend to expand our analysis to include spatial and temporal analysis to find out when and where crime has spread in the past, and when and where it is most likely to spread in the future.

2.3 SOFTWARE REQUIREMENTS

- Operating system: Windows 7.
- Coding Language: python
- Tool: anaconda, visual studio code
- Libraries: OpenCV

2.4 HARDWARE REQUIREMENTS

- System : Pentium i3 Processor.
- Hard Disk : 500 GB.
- Monitor : 15" LED

- Input Devices : Keyboard, Mouse, Ram : 2 GB

CHAPTER – 3

METHODS AND ALGORITHMS USED

3.1 DATA COLLECTION

- Social media posts are collected through the Search API .
- The search of the social media posts must be based on a set of keywords that can be used to classify the crime situations.
- Thus in the first filter, we used the main crime-related keywords according to crime categories.

3.2 DATA CLEANING

- Data cleaning is a critically important step in any machine learning project.
- Data Cleaning is done to prepare the data for analysis by removing or modifying the data that may be incorrect, incomplete, duplicated or improperly formatted.
- There are many different statistical analysis and data visualization techniques you can use to explore your data in order to identify data cleaning operations you may want to perform.

3.3 DATA PREPROCESSING

- As in fig 3.1, it is very important to apply the pre- processing techniques to the extracted data set.
- Because, there may be typos, unwanted content like URLs, and stop words in the social media post.
- Thus, data which is obtained from social media is highly unstructured and noisy.

- Pre-processing techniques will generate clean tweet data that will be used for the next process.
- First, we removed the stop words such as is, the, which, have, etc. The words do not convey any positive or negative meaning. So, we can easily remove the stop word without affecting the meaning of the message.
- Then, we removed URLs, hashtags, symbols, usernames, expressions, quotes, etc. Next, combine words are split by applying tokenizing techniques.
- Finally, we applied a stemming algorithm to reduce a word to its word stem that affixes to suffixes and prefixes or the roots of words.

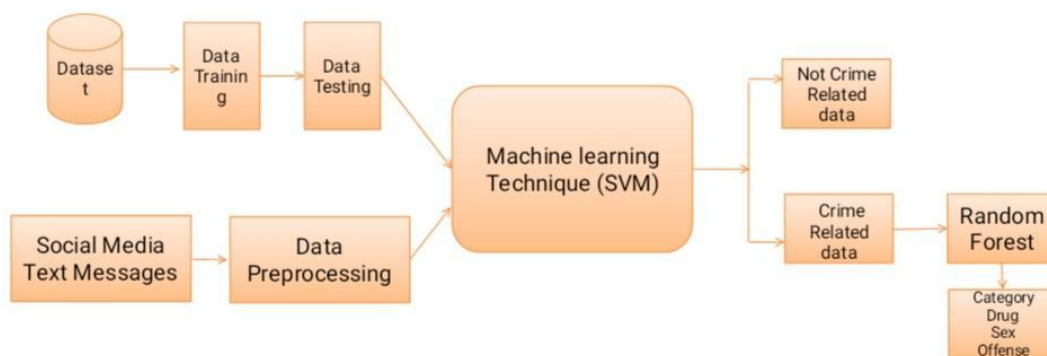


Fig 3.1 System Architecture

3.4 FEATURE EXTRACTION

- Feature Extraction is done to reduce the number of attributes in the dataset hence providing advantages like speeding up the training and accuracy improvements.
- In machine learning, pattern recognition, and image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

- When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector).
- Determining a subset of the initial features is called feature selection. The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

3.5 DATA PREPERATION

- After completing the pre-processing, social media posts are transformed into vectors to generate the feature vectors. The vectors are used in the learning phase for machine learning algorithms.
- It is used as the term frequency- inverse document frequency (TF-IDF) values to create the vectors.
- TF-IDF value reflects the importance of a term in a document to the collection of documents.

3.6 SECOND STAGE FILTERING

- Despite a large amount of data collected via keyword filtering, few of them are concerning crimes. Most information is considered as noise.
- According to the above examples, the searched keyword is 'shoot'. We have expected gun shoot data relates to crime scenes. However, the above post is related to a film shoot and photo shoot. So, we can consider it as a noisy post.
- To eliminate the noise and increase the accuracy, we implemented a machine learning-based filter in this stage. We have used the Support Vector Machine (SVM) as the machine learning algorithm.
- The SVM is a state-of-the-art classification method that uses a learning algorithm based on structural risk minimization.

- The classifier can be used in many disciplines because of its high accuracy, ability to deal with high dimensions, and flexibility in modeling diverse sources of data.

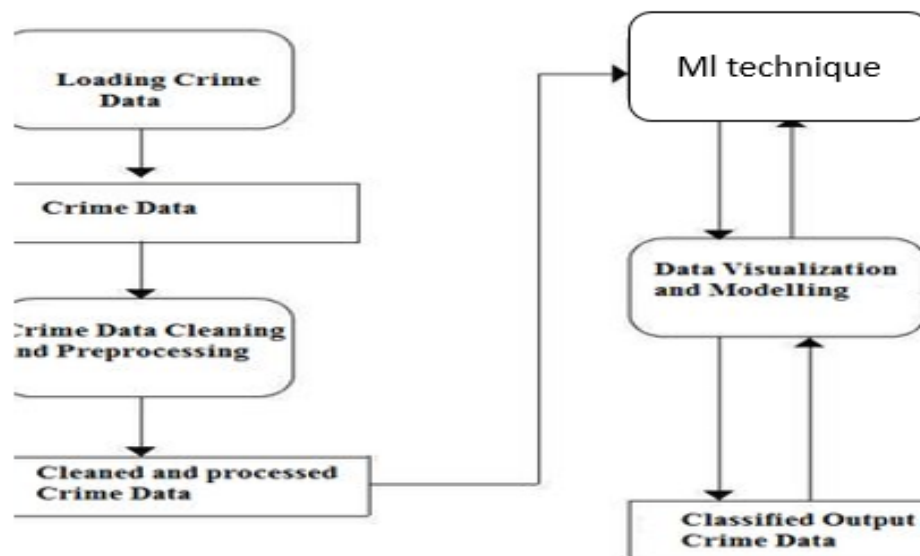


Fig 3.2 Block Diagram

- As in fig 3.2, It represents the block diagram which predicts the following techniques, such as ML technique, data visualization, and it gives classified output based on the following data processing.

3.7 Testing model

- In this module we test the trained machine learning model using the test dataset
- Quality assurance is required to make sure that the software system works according to the requirements. Were all the features implemented as agreed? Does the program behave as expected? All the parameters that you test the program against should be stated in the technical specification document.
- Moreover, software testing has the power to point out all the defects and flaws during development. You don't want your clients to encounter bugs

after the software is released and come to you waving their fists. Different kinds of testing allow us to catch bugs that are visible only during runtime.

3.8 Performance Evaluation

- In this module, we evaluate the performance of trained machine learning model using performance evaluation criteria such as F1 score, accuracy and classification error.
- In case the model performs poorly, we optimize the machine learning algorithms to improve the performance.
- performance Evaluation is defined as a formal and productive procedure to measure an employee's work and results based on their job responsibilities. It is used to gauge the amount of value added by an employee in terms of increased business revenue, in comparison to industry standards and overall employee return on investment (ROI).
- All organizations that have learned the art of "winning from within" by focusing inward towards their employees, rely on a systematic performance evaluation process to measure and evaluate employee performance regularly.
- Ideally, employees are graded annually on their work anniversaries based on which they are either promoted or are given suitable distribution of salary raises.
- Performance evaluation also plays a direct role in providing periodic feedback to employees, such that they are more self-aware in terms of their performance metrics.

3.9 Prediction

- Prediction" refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days.

- The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be.
- The word “prediction” can be misleading. In some cases, it really does mean that you are predicting a future outcome, such as when you’re using machine learning to determine the next best action in a marketing campaign.
- Other times, though, the “prediction” has to do with, for example, whether or not a transaction that already occurred was fraudulent.
- In that case, the transaction already happened, but you’re making an educated guess about whether or not it was legitimate, allowing you to take the appropriate action.
- In this module we use trained and optimized machine learning model to predict whether the crime accrue in social media.

3.9.1 NLTK

- NLTK is a toolkit build for working with NLP in Python.
- It provides us various text processing libraries with a lot of test datasets.
- A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc
- Tokenization
- Lower case conversion
- Stop Words removal
- Stemming
- Lemmatization
- Parse tree or Syntax Tree generation
- POS Tagging

3.9.2 TensorFlow

- TensorFlow is an open source framework developed by Google researchers to run machine learning, deep learning and other statistical and predictive analytics workloads. The TensorFlow software handles data sets that are arrayed as computational nodes in graph.
- It is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation

3.9.3 KERAS

- Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.
- Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.
- Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.
- In fig 3.3, Here the work flow diagram represents the following methods which is involved in the process of crime prediction, such as data collection, data testing, data preprocessing and using of support vector machine and random forest algorithms to detect and analyze the crime related data from dataset.

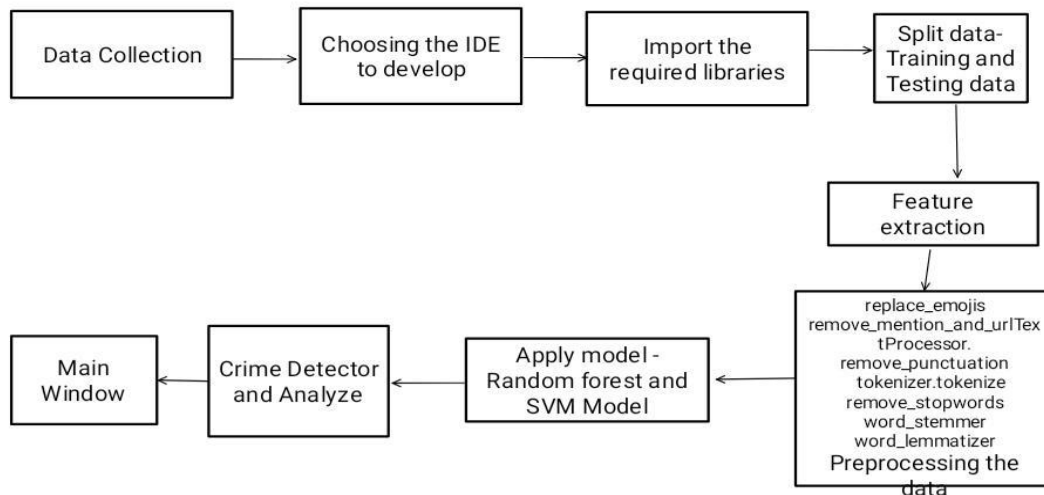


Fig 3.3 Work Flow Diagram

3.10 Support Vector Machine (SVM) algorithm

Mastering machine learning algorithms isn't a myth at all. Most of the beginners start by learning regression. It is simple to learn and use, but does that solve our purpose? Of course not! Because you can do so much more than just Regression!

Think of machine learning algorithms as an armory packed with axes, sword, blades, bow, dagger, etc. You have various tools, but you ought to learn to use them at the right time. As an analogy, think of 'Regression' as a sword capable of slicing and dicing data efficiently, but incapable of dealing with highly complex data. On the contrary, 'Support Vector Machines' is like a sharp knife – it works on smaller datasets, but on the complex ones, it can be much stronger and powerful in building machine learning models.

By now, mastered Random Forest, Naive Bayes Algorithm and Ensemble Modeling. If not, I'd suggest you take out a few minutes and read about them as well. In this article, I shall guide you through the basics to advanced knowledge of a crucial machine learning algorithm, support vector machines.

You can learn about Support Vector Machines in course format here (it's free!):

If you're a beginner looking to start your data science journey, you've come to the right place! Check out the below comprehensive courses, curated by industry experts, that we have created just for you:

3.10.1 What is Support Vector Machine?

Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, as shown in fig 3.4, we plot each data item as a point in n - dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

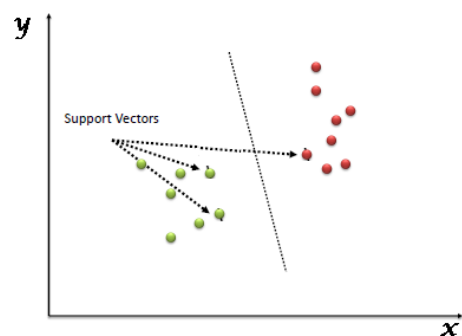


Fig 3.4 N – Dimensional Space

Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

You can look at support vector machines and a few examples of its working here.

3.10.2 How does it work?

Above, we got accustomed to the process of segregating the two classes with a hyper-plane. Now the burning question is "How can we identify the right hyper-plane?". Don't worry, it's not as hard as you think!

Let's understand

Identify the right hyper-plane (Scenario-1): Here, as in fig 3.5 we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.

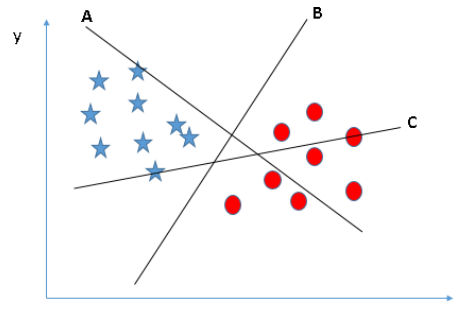


Fig 3.5 Hyper-Planes A, B and C

You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

Identify the right hyper-plane (Scenario-2): Here, in fig 3.6 have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

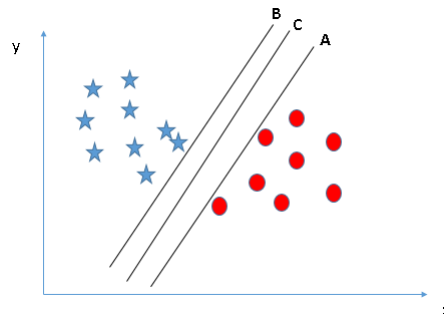


Fig 3.6 Hyper-Planes Segregation

Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin. Let’s look at the below snapshot as in fig 3.7.

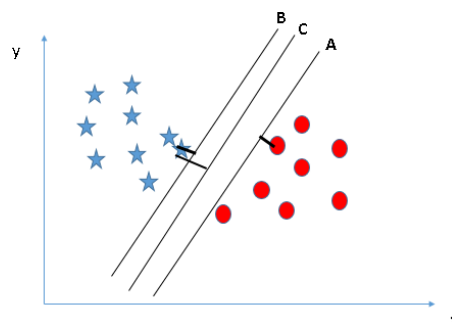


Fig 3.7 Maximizing the distances

Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

Identify the right hyper-plane (Scenario-3) Hint: Use the rules as discussed in previous section to identify the right hyper-plane as shown in fig 3.8.

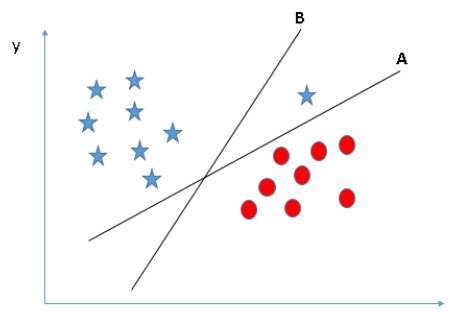


Fig 3.8 Identifying the Hyper-Plane

Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A**. But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A**.

Can we classify two classes (Scenario-4)? Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier(fig 3.9).



Fig 3.9 Classifying the classes

As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.

Find the hyper-plane to segregate to classes (Scenario-5)

In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.

SVM can solve this problem. Easily! It solves this problem by introducing additional feature(fig 3.10). Here, we will add a new feature $z=x^2+y^2$. Now, let's plot the data points on axis x and z:

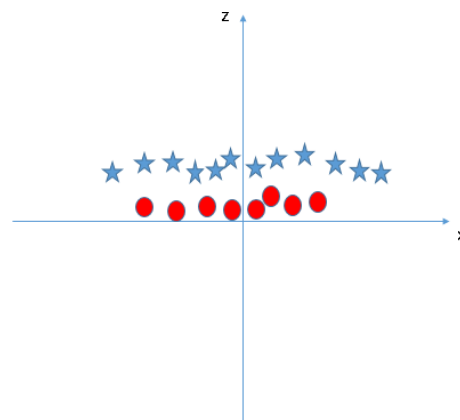


Fig 3.10 Additional Feature

In above plot, points to consider are:

All values for z would be positive always because z is the squared sum of both x and y

In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the kernel **trick**. The SVM kernel is a

function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

When we look at the hyper-plane in original input space it looks like a circle as in fig 3.11.

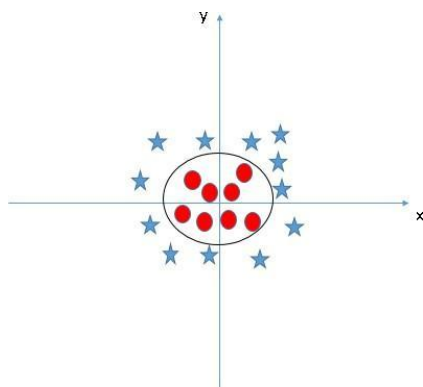


Fig 3.11 Original Input Space

Now, let's look at the methods to apply SVM classifier algorithm in a data science challenge.

Problem Statement:

Dream Housing Finance company deals in all home loans. They have a presence across all urban, semi-urban and rural areas. A customer first applies for a home loan, after that the company validates the customer's eligibility for a loan.

Company wants to automate the loan eligibility process (real-time) based on customer details provided while filling an online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers' segments, those are eligible for loan amount so that they can specifically target these customers. Here they have provided a partial data set.

Pros

- It works really well with a clear margin of separation. It is effective in high dimensional spaces.
- It is effective in cases where the number of dimensions is greater than the number of samples.
- It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Cons

- It doesn't perform well when we have large data set because the required training time is higher
- It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
- SVM doesn't directly provide probability estimates, these are calculated using an expensive five- fold cross-validation. It is included in the related SVC method of Python scikit-learn library.

3.11 Random Forest Algorithm

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.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Real Life Analogy

Let's dive into a real-life analogy to understand this concept further. A student named X wants to choose a course after his 10+2, and he is confused about the choice of course based on his skill set. So he decides to consult various people like his cousins, teachers, parents, degree students, and working people. He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc. Finally, after consulting various people about the course he decides to take the course suggested by most of the people.

3.11.1 Working of Random Forest Algorithm

Before understanding the working of the random forest we must look into the ensemble technique. **Ensemble** simply means combining multiple models. A collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

1. **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
2. **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST

As mentioned earlier, Random forest works on the Bagging principle. Now let's dive in and understand bagging in detail.

Bagging

Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row

sampling with replacement is called bootstrap (fig 3.12). Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This

step which involves combining all the results and generating output based on majority voting is known as aggregation.

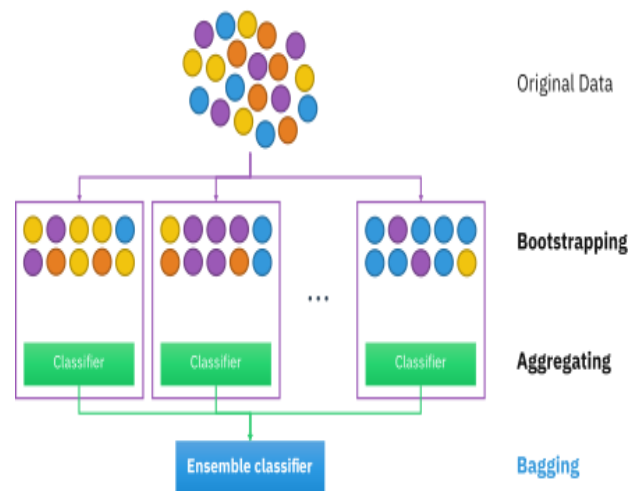


Fig 3.12 Bootstrapping

Now let's look at an example by breaking it down with the help of the following figure. Here the bootstrap sample is taken from actual data (Bootstrap sample 01, Bootstrap sample 02, and Bootstrap sample 03) with a replacement which means there is a high possibility that each sample won't contain unique data.

Now the model (Model 01, Model 02, and Model 03) obtained from this bootstrap sample is trained independently. Each model generates results as shown. Now Happy emoji is having a majority when compared to sad emoji. Thus based on majority voting final output is obtained as Happy emoji

Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

For example: consider the fruit basket as the data as shown in the figure below. Now n number of samples are taken from the fruit basket and an

individual decision tree is constructed for each sample. Each decision tree will generate an output as shown in the figure. The final output is considered based on majority voting. In the below figure you can see that the majority.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 RESULTS

Thus, the program for CRIME PREDICION using “MACHINE LEARNING” has been executed successfully. The execution is done in two parts i.e, testing and training. Training is the phase where the dataset is trained continuously so that it Detects the Crime related words and Analyses the crime rate in the testing phase and gives output as per prediction and condition.

4.2 SCREENSHOTS

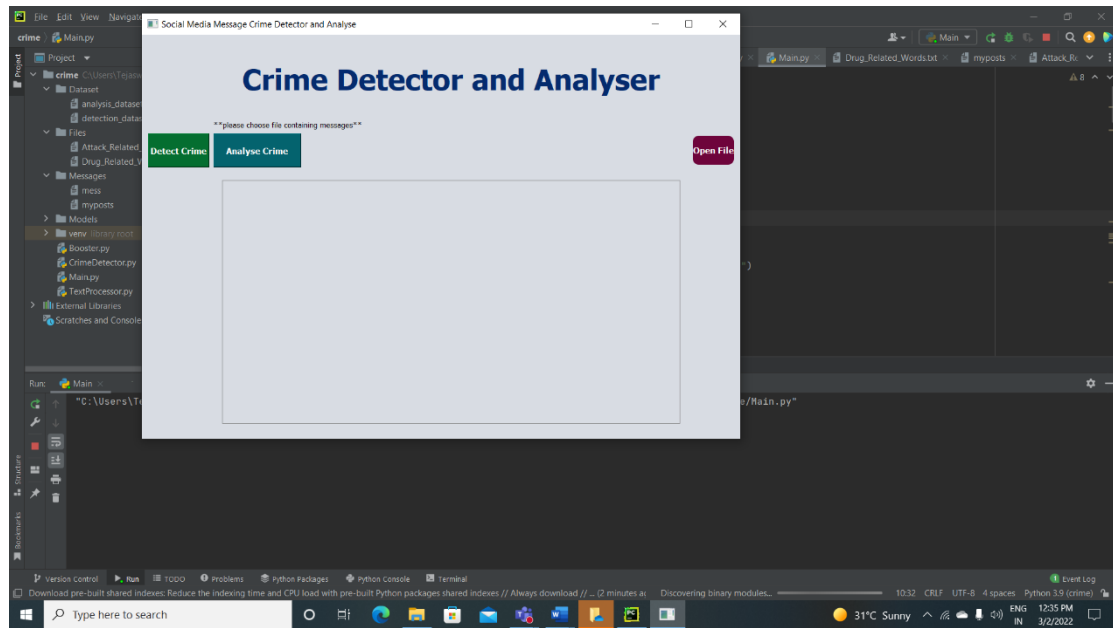


Fig 4.1 Screen Shot of OUpot

In the fig 4.1, Here we can see a picture which is representing the Detector and which is analyzing the crime, so here we can click on open file then we can browse a dataset.

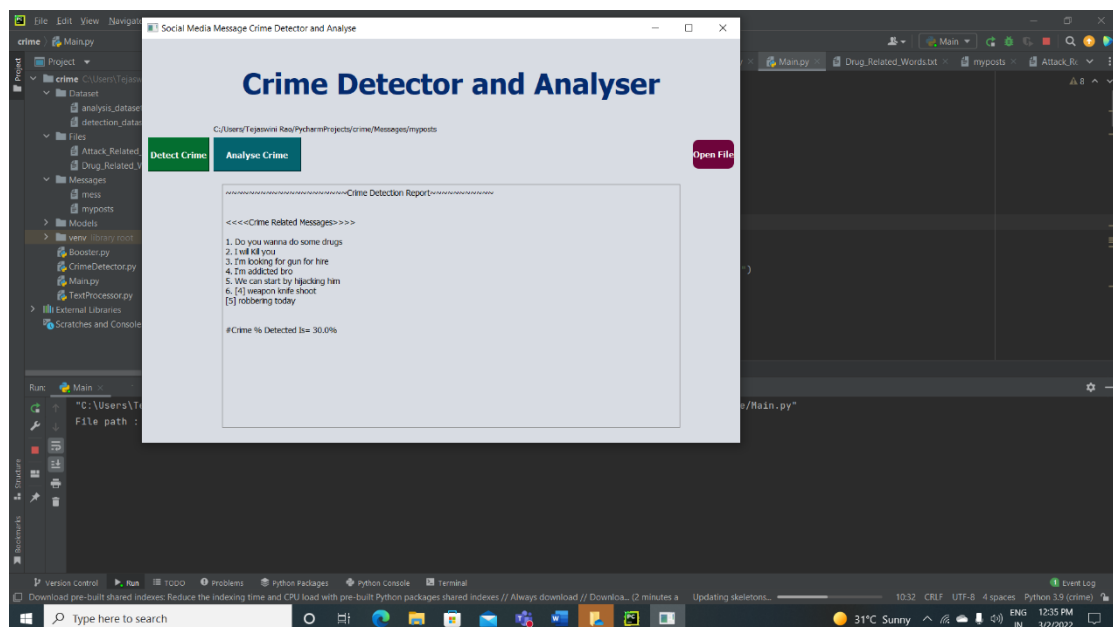


Fig 4.2 Crime Detection Report

In the fig 4.2, shows that, after the input is given then it is detected into crime related and non-crime related messages by SVM algorithm. And, finally it shows crime related messages as output.

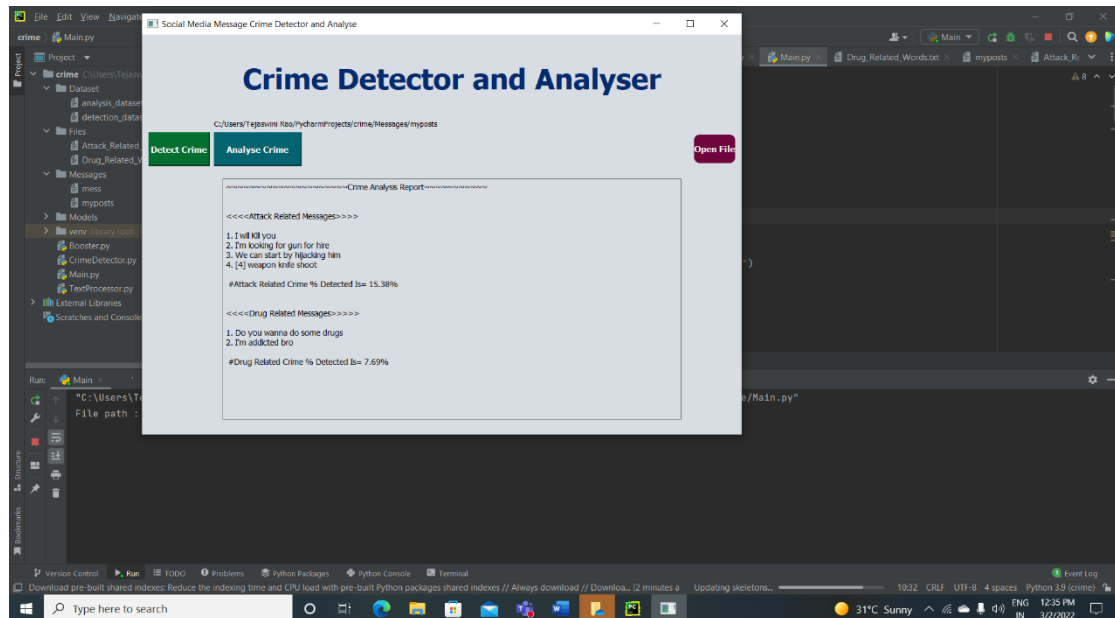


Fig 4.3 Crime Analysis Report

In the fig 4.3, shows that, after detecting the output it is further analyzed into crime types by using random forest algorithm. And it displays in the output as it is.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

we proposed an approach to improving the accuracy of the detection of crime-related posts from social media text messages. Here, first, we applied a keyword-based filter, and then to remove the noise, we applied the SVM based filter and random forest classification. According to the existing works, SVM has the best accuracy among the classifiers. So, we used SVM for the research.

5.2 FUTURE ENHANCEMENT

Though it overcomes the problem of the existing work, it has some limitations. In the situation of absence of class labels, then the probability of the estimation will be zero. As a future extension of the proposed work, the application of more machine learning classification models proves to increase accuracy in crime prediction and will enhance the overall performance. It helps in providing a better study for the future improvement by taking the income information into consideration for neighborhoods places in order to foresee if any relationship between the income levels of a particular in the neighborhood places and their crime rate.

REFERENCES

JOURNALS

- [1] P. Sivakumar, J. V, R. R and K. S, "Real Time Crime Detection Using Deep Learning Algorithm," 20[21 International Conference on System, Computation, Automation and Networking (ICSCAN),2021,pp.1-5, doi: 10.1109/ICSCAN53069.2021.9526393.
- [2] H. Farhood, M. Saberi and M. Najafi, "Improving Object Recognition in Crime Scenes via Local Interpretable Model-Agnostic Explanations," 2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW), 2021, pp. 90-94, doi: 10.1109/EDOCW52865.2021.00037.
- [3] A. A. Almuhanha, M. M. Alrehili, S. H. Alsubhi and L. Syed, "Prediction of Crime in Neighbourhoodsof New York City using Spatial Data Analysis," 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), 2021, pp. 23-30, doi: 10.1109/CAIDA51941.2021.9425120.
- [4] S. P. C. W. Sandagiri, B. T. G. S. Kumara and B. Kuhaneswaran, "ANN Based Crime Detection and Prediction using Twitter Posts and Weather Data," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-5, doi: 10.1109/ICDABI51230.2020.9325660.
- [5] A.-S. Hissah and H. Al-Dossari, "Detecting and Classifying Crimes from Arabic Twitter Posts using Text Mining Techniques," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 9, 2018
- [6] J. Bendler, A. Ratku, and D. Neumann, "Crime mapping through geo-spatial social media activity," 2018.
- [7] L. E. Cohen and M. Felson, "Social change and crime rate trends: A routine activity approach," American sociological review, pp. 588-608, 2017.

[8] Benjamin Fredrick David. H and A. Suruliandi, "Survey on Crime Analysis and Prediction using Data mining techniques", ICTACT Journal on Soft Computing on April 2012.

[9] Learning Rohit Patel, Muzamil Kacchi, Pranali Gavali and Komal Pimparia, "Crime pattern detection ,analysis & prediction using machine", International research Journal of Engineering and Technology, (IRJET) e-ISSN: 2395-0056, Volume: 07, Issue: 06, June 2020.

APPENDIX

A. SOURCE CODE

```
import sys

from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *

from Booster import DetectorBooster, AnalyzorBooster


class OtherWindow(QMainWindow):
    def __init__(self, parent=None):
        super().__init__()
        self.setWindowTitle("Social Media Message Crime Detector and Analyse")
        self.setGeometry(100, 50, 400, 600)
        self.show()
        self.detection_booster = None

        self.analysis_booster = None
        self.is_file_chosen = False


class MainWindow(QMainWindow):
    def __init__(self, parent=None):
        super().__init__()
        self.setWindowTitle("Social Media Message Crime Detector and Analyse")

        btn = QPushButton("Open File")
```

```

self.report_txt = QTextEdit()
self.report_txt.setStyleSheet("font-size: 14px; margin:15; border-color:
#042b6e ;border-width: 2px")
btn.setStyleSheet("background-color: #6e043b;"
" color:white;font-weight:bold ;"
"font-size:15px; max-height:
50px;max-width: 150px; border-radius:10px")
detect_btn = QPushButton("Detect Crime")
detect_btn.setStyleSheet("background-color: #046e30;
color:white;font-weight:bold ;"
"font-size:15px;height: 50px;max-width: 150px")
analyse_btn = QPushButton("Analyse Crime")
analyse_btn.setStyleSheet("background-color:
#04636e;color:white;font- weight:bold ;"
"font-size:15px;height: 50px;max-width: 150px;"
header = QLabel("Crime Detector and Analyser")
header.setFont(QFont('Times', 30))
header.setStyleSheet("font-weight: bold;color: #042b6e; margin:35")

self.file_name_lbl = QLabel("***please choose file containing messages**")

layout = QGridLayout()
layout.addWidget(header, 1, 1)
layout.addWidget(self.file_name_lbl, 2, 1)
layout.addWidget(btn, 3, 3)
layout.addWidget(detect_btn, 3, 0)
layout.addWidget(analyse_btn, 3, 1)
layout.addWidget(self.report_txt, 4, 1)

widget = QWidget()
widget.setAttribute(Qt.WA_StyledBackground, True)

```

```

widget.setStyleSheet('background-color: #d8dce3;')

widget.setLayout(layout)
self.setCentralWidget(widget)

btn.clicked.connect(self.openFile) # connect clicked to self.open()
detect_btn.clicked.connect(self.printDetectionReport)
analyse_btn.clicked.connect(self.printAnalysisReport)

self.setGeometry(230, 50, 500, 700)
self.show()

def openFile(self):
    path = QFileDialog.getOpenFileName(self, 'Open a file', "",
                                      'All Files (*.*)')
    if path != ("", ""):
        print("File path : " + path[0])
        self.file_name_lbl.setText(path[0])
        self.detection_booster = DetectorBooster(filename=path[0])
        self.analysis_booster = AnalyzorBooster(filename=path[0])
        self.is_file_chosen = True

def printDetectionReport(self):

    if self.is_file_chosen:
        num = 1
        report = "~~~~~Crime

Detection Report~~~~~\n\n\n"
        crime_list, rate = self.detection_booster.detect()

        report += "<<<<Crime Related Messages>>>>\n\n"
        for m in crime_list:
            report += str(num) + ". " + m + "\n"

```

```

num += 1
rate = str(round(rate * 100, 2))

report += "\n\n#Crime % Detected Is= " + rate + "%"

self.report_txt.setPlainText(report)
else:
self.report_txt.setPlainText("*****Choose file first*****")

def printAnalysisReport(self):
If self.is_file_chosen:
report = "~~~~~Crime

Analysis Report~~~~~\n\n\n"
attack_list, drug_list, a_rate, d_rate = self.analysis_booster.analyse()

num = 1
report += "<<<<Attack Related Messages>>>>\n\n"
for m in attack_list:

report += str(num) + ". " + m + "\n"
num += 1
a_rate = str(round(a_rate * 100, 2))
report += "\n #Attack Related Crime % Detected Is= " + a_rate + "%\n\n\n"

num = 1
report += "<<<<Drug Related Messages>>>>\n\n"
for m in drug_list:

report += str(num) + ". " + m + "\n"
num += 1
d_rate = str(round(d_rate * 100, 2))
report += "\n #Drug Related Crime % Detected Is= " + d_rate + "%"

self.report_txt.setPlainText(report)
else:

```

```
self.report_txt.setPlainText("*****Choose file first*****")
```

```
if __name__ == "__main__":  
    app = QApplication(sys.argv)  
    window = MainWindow()  
    sys.exit(app.exec_())
```