A Project Report

On

"CYBER SAVER- A MACHINE LEARNING APPROACH TO DETECTION OF CYBERBULLYING"

Submitted in partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND DESIGN

Submitted by

T. TRISHA	**(213N1A3862)**
K. RAGUHNATH	**(213N1A3833)**
M. SUMANTH	**(213N1A3841)**
G. LIKITHA	**(213N1A3820)**

Under the esteemed guidance of

Mrs. R. SRAVANI M.Tech,(Ph.D)

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND DESGIN

P.V.K.K Institute of Technology

An ISO 9001-2015 Certified Institute

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu.)

Sanapa Road, Rudrampeta, Anantapuramu-515001

Andhra Pradesh

2024-2025

P.V.K.K INSTITUTE OF TECHNOLOGY

An ISO 9001-2015 Certified Institute
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)
Sanapa road, Rudranpeta, Anantapuramu-515001

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

CERTIFICATE

This is to certify that project report entitled **"CYBER SAVER – A MACHINE LEARNING APPROACH TO DETECTION OF CYBER BULLYING"** is bonafide work done.

by

T. TRISHA	**(213N1A3862)**
K. RAGUHNATH	**(213N1A3833)**
M. SUMANTH	**(213N1A3841)**
G. LIKITHA	**(213N1A3820)**

Under the supervision of **R. SRAVANI** M.Tech,(Ph.D) Assistant Professor. of CSE in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** with specialization in **COMPUTER SCIENCE AND DESIGN** in **P.V.K.K Institute of Technology**, Sanapa Road, Rudrampeta, Anantapuramu during the academic year 2024-2025.

Signature of the Project Guide
R. SRAVANI, M. Tech, (Ph. D)
Assistant Professor
Dept. of CSE

Signature of HOD
Dr. Dilip Venkata Kumar, M. Tech, Ph. D
Professor & HOD
Dept. of CSG

External Viva-voce Held on:______________________

External Examiner

DECLARATION

We hereby declare that the project work entitled "**Cyber Saver – A Machine Learning Approach to Detection of Cyber Bullying**" is a genuine work carried out by us under the guidance of **Mrs. R. Sravani M.Tech,(Ph.D) Asst. Professor, Department of Computer Science & Engineering**, in partial fulfillment for the award of the degree of **"B.Tech"** of **Jawaharlal Nehru Technological University Anantapur, Anantapuramu**. This has not been submitted in part of fulfillment towards any other degree.

T. TRISHA	K. RAGUHNATH
213N1A3862	213N1A3833

M. SUMANTH	G. LIKITHA
213N1A3841	213N1A3820

ACKNOWLEDGEMENT

At the outset, we thank our Honourable Founder, Correspondent & Secretary **Dr. PALLE RAGHUNATHA REDDY** M.Sc.,M.Phil,Ph.D.,garu, Honourable Chairman **Dr. PALLE VENKATA KRISHNA KISHORE REDDY** M.Tech, MS(USA),AMIE...Ph.D., garu, and our Treasurer Smt**. P. SINDHURA REDDY**M.Tech**.**, garu of Sri Balaji Educational Society for providing us with good faculty and all the required equipment in C.S.E labs and for their moral support throughout the course.

I wish to express our sense of gratitude to **Dr. B. RAMESH BABU** B. Tech, M.S, Ph.D., garu Principal, **P.V.K.K INSTITUTE OF TECHNOLOGY**, for providing necessary facilities.

With a great sense of pleasure, we extend our gratitude to **Dr. DILIP VENKATA KUMAR.** M.Tech, Ph.D., garu, Head of Computer Science & Design for his co-operation and providing necessary help for completion of this project.

It is a great pleasure to express my heartful gratitude and thankfulness to my guide Mrs. **R. SRAVANI** M.Tech**,**(Ph.D) garu, Assistant Professor, Department of Computer Science& Engineering for her valuable guidance and helpful counsel rendered in the due course of the project.

Finally, we would like to thank one and all in the C.S.E department who helped us directly and indirectly along with my family members and friends for their cooperation to complete this project.

PROJECTASSOCIATES

T. TRISHA

K. RAGUHNATH

M. SUMANTH

G. LIKITHA

TABLE OF CONTENTS

ABSTRACT

In this modern era of extensive use of online resources there have been reports of numerous cases of cyber bullying. Although awareness through medical health support systems such as counselling and psychological assistance is available, a system to combat threats is needed to handle the increasing rate of cyber bullying. This paper presents a model that can be used to detect and report cyber bullying with the use of machine learning techniques. A careful selection of the machine learning algorithms has been identified that could enable better accurate detection. The model was transformed into a prototype in python to evaluate the effectiveness of the model in detecting cyber bullying. The proposed model primarily focuses on test based and image-based threats as they are more common than other forms of cyber bullying.

LIST OF FIGURES

LIST OF ABBREVIATION

S. No	Short Form	Abbreviation
1	AI	Artificial Intelligence
2	ML	Machine Learning
3	NLP	Natural Language Processing
4	SVM	Support Vector Machine
5	NB	Naïve Bayes
6	RF	Random Forest
7	LR	Logistic Regression
8	KNN	K-Nearest Neighbours
9	CNN	Convolutional Neural Network
10	RNN	Recurrent Neural Network
11	LSTM	Long Short-Term Memory
12	TF-IDF	Term Frequency-Inverse Document Frequency
13	BOW	Bag of Words
14	ROC	Receiver Operating Characteristic
15	AUC	Area Under the Curve
16	F1	F1 Score (harmonic mean of precision and recall)
17	CSV	Comma-Separated Values
18	GBT	Gradient Boosting Tree
19	SKLEARN	Scikit-Learn (Python ML library)
20	NLTK	Natural Language Toolkit

CHAPTER-1

INTRODUCTION

Social media networks such as Face book, Twitter, Flickr, and Instagram have become the preferred online platforms for interaction and socialization among people of all ages. While these platforms enable people to communicate and interact in previously unthinkable ways, they have also led to malevolent activities such as cyber-bullying. Cyber bullying is a type of psychological abuse with a significant impact on society. Cyber-bullying events have been increasing mostly among young people spending most of their time navigating between different social media platforms. Particularly, social media networks such as Twitter and Face book are prone to CB because of their popularity and the anonymity that the Internet provides to abusers. In India, for example, 14 percent of all harassment occurs on Face book and Twitter, with 37 percent of these incidents involving youngsters [1]. Moreover, cyber bullying might lead to serious mental issues and adverse mental health effects. Most suicides are due to the anxiety, depression, stress, and social and emotional difficulties from cyber-bullying events [2][4]. This motivates the need for an approach to identify cyber bullying in social media messages (e.g., posts, tweets, and comments) we mainly focus on the problem of cyber bullying detection on the Twitter platform. As cyber bullying is becoming a prevalent problem in Twitter, the detection of cyber bullying events from tweets and provisioning preventive measures are the primary tasks in battling cyber bullying threats [5]. Therefore, there is a greater need to increase the research on social networks-based in order to get greater insights and aid in the development of effective tools and approaches to effectively combat cyber bullying problem [6]. Manually monitoring and controlling cyber bullying on Twitter platform is virtually impossible [7]. Furthermore, mining social media messages for cyber bullying detection is quite difficult. For example, Twitter messages are often brief, full of slang, and may include emojis, and gifs, which makes it impossible to deduce individuals' intentions and meanings purely from social media messages. Moreover, bullying can be difficult to detect if the bully uses strategies like sarcasm or passive-aggressiveness to conceal it. Despite the challenges that social media messages bring, cyber bullying detection on social media is an open and active research topic. Cyber bullying detection within the Twitter platform has largely been pursued through tweet classification and to a certain extent with topic modelling approaches. Text classification based on supervised machine learning These models help in leveraging the bidirectional processing.

To extract meaningful topics. However, these unsupervised models require extensive training to obtain sufficient prior knowledge, which is not adequate in all cases [25]. Considering these limitations, an efficient tweet classification approach must be developed to bridge the gap between the classifier and the topic model so that the adaptability is significantly proficient.

MOTIVATION

The surge in cyberbullying amid widespread online activity necessitates proactive measures. While counselling and psychological support address aftermaths, a system to counteract threats is crucial. This project's motivation lies in developing a machine learning-based model to accurately detect and report cyberbullying, specifically targeting text and image-based threats for a more comprehensive solution.

1.1 PROBLEM DEFINITION

The problem addressed in this project is the escalating prevalence of cyberbullying in the modern era. Despite available support systems like counselling, there's a need for a proactive solution to combat the rising threat. The project aims to develop a machine learning-based model focusing on detecting and reporting text and image-based cyber bullying incidents effectively.

1.2 OBJECTIVE OF PROJECT

The primary objective of this project is to develop a machine learning-based model for the detection and reporting of cyber bullying, with a specific focus on text and image-based threats. The goal is to provide a proactive and accurate solution to mitigate the increasing instances of cyber bullying in the online environment.

1.3 OVERVIEW OF THE PROJECT

This project aims to develop a machine learning-based model for the detection and reporting of cyberbullying on social media platforms, focusing on both text and image-based threats. The rise of platforms like Twitter, Facebook, and Instagram has made online interaction easier, but it has also led to a significant increase in cyberbullying, which can have severe mental health consequences, particularly among young people. Detecting cyberbullying is challenging due to the nature of social media messages, which often include slang, emojis, gifs, and subtle forms of communication like sarcasm or passive-aggressive language. Current methods, such as supervised machine learning and topic modelling, are limited by their need for large datasets and difficulty in adapting to new forms of bullying, motivating the need for a more adaptable and effective detection system.

CHAPTER-2

LITERATURE SURVEY

1. M. Di Capua, et al. [1] raises the unsupervised how to develop an online bullying model based on a combination of features, based on traditional textual elements and other "social features". Features were divided into 4 categories as Syntactic features, Semantic features, Sentiment features, and Community features. The author has used the Growing Hierarchical Self Map editing network (GHSOM), with 50 x grid 50 neurons and 20 elements as the insertion layer. M. Di Capua, and others used an integration algorithm k-means to separate the input database and GHSOM in the Formspring database. The effects of this hybrid the unsupervised way surpasses the previous one results. The author then checked the youtube database at 3 p.m. Different Machine Learning Models: Naive Bayes Classifier, Decision Tree Classifier (C4.5), and Support Vector Machine (SVM) with Linear Kernel. It was saw that the combined effects of hate speech turned around so that we have lower accuracy in the youtube database compared to FormSpring tests, as in the text analysis and syntactical features work differently in on both sides. When this hybrid method is used in Twitter Database, led to weak memory and F1 Score. The model proposed by the authors can also be improved used in building constructive mitigation applications cyberbullying problems.

2. J. Yadav, et al. [2] suggests a new approach to this the discovery of internet cyberbullying on social media in using a BERT model with a single line neural and was tested on the Formspring forum and Wikipedia Database. The proposed model provided performance 98% accuracy of spring Form databases and 96% accuracy in a relatively comprehensive Wikipedia database previously used models. The proposed model provided better Wikipedia database results due to its size g without the need for excessive sampling while I The spring data form requires multiple samples.

3. R. R. Dalvi, et al. [3] suggests a way to do this detect and prevent online exploitation on Twitter using Classified supervised machine learning algorithms. In this study, the live Twitter API is used for compilation tweets and data sets. The proposed model tests both Support Vector Machine and Naive Bayes on the data sets are collected. To remove a feature, use it TFIDF vectorizer.

The results show that it is accurate of an online bullying model based on Vector Support The machine is about 71.25% better than Naive Bayes was almost 52.75%.

4. Trana R.E., et al. [4] The goal was to design a machine learning model to minimize special events including text extracted from image memes. Author include asite that contains approximately 19,000 text views have been published on YouTube. This study discusses the operation of three learning machines equipment, Uninformed Bayes, Support Vector The machine, as well as the convolutional neural network used in on the YouTube website, and compare the results with existing details of the Form. They do not write continuously investigate cyber bullying algorithms sub-sections within the YouTube website. Naive Bayes beat SVM and CNN in the next four categories: race, nationality, politics, and general. SVM passed well with the inexperienced Naïve Bayes again CNN is in the same gender group, with all three algorithms show equal performance with the middle body group accuracy. The results of this study provided inaccurate data used to distinguish between incidents of abuse and non-violence. Future work can focus on the construction of a two-part separation system used to test text taken from photos to see that the YouTube website provides a better context for aggression related collections.

5. N. Tsapatsoulis, et al. [5] detailed review of introduced cyberbullying on Twitter. The importance of identifying the various abusers on Twitter is provided. Kwe paper, various practical steps required the development of an effective and efficient application for Internet traffic detection is well defined styles involved in data classification and recording platforms, machine learning models and feature types, and model studies using such tools explained. This paper will serve as the first step in the process project on acquisition of cyberbullying technology using the machine reading.

6. G. A. León-Paredes et al. [6] they described I the development of an online bullying detection model is used Native Language Processing (NLP) and Mechanics Reading (ML). Spanish Cyberbullying Prevention The system (SPC) was developed by installing the machine learning strategies Naïve Bayes, Support Vector Machine, and Logistic Regression. Database used this study was posted on Twitter. Plurality 93% accuracy was achieved with the help of third parties techniques used. Charges of online exploitation were found with the help of this system presented the accuracy of 80% to 91% on average. Stemming and lemmatization techniques in NLP can be used continuously increasing system accuracy. Such a model can and used for adoption in English and local languages if possible.

7. P. K. Roy, et al. [7] details about creating a request for hate speech on Twitter via the help of a deep neural convolutional network. Machine learning algorithms such as Logistic Regression (LR), Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Gradient Boosting (GB), and K-nearby Neighbors (KNN) have it used to identify tweets related to hate speech in Twitter and features have been removed using tf-idf process. The leading ML model was SVM but it managed predict 53% hate speech tweets on a 3: 1 database to be tested train. The reason behind the low forecast scale was unequal data. The model is based on the prediction of hate speech tweets. Advanced learning-based methods on the have the same effects as separate distributed database. 10 times cross confirmation was used in conjunction with the proposed once you got a very good rate of memory. It was 0.88 hate speech and 0.99 non-hate speech test results confirmed that the k-fold opposite verification process is a better resolution with unequal data. In the future, the current database can be expanded for better performance accuracy.

In conclusion, the advancements in machine learning techniques have significantly contributed to the detection and prevention of cyberbullying across social media platforms. The various models explored in the literature, such as the hybrid approach by M.Di Capua et al. [1] and the BERT model by J. Yadav et al. [2], highlight the potential of integrating traditional and advanced machine learning methods to tackle the challenges of detecting harmful online behaviour. The hybrid approach that combines textual and social features has shown promising results, though its performance varies across platforms, emphasizing the need for platform-specific adaptations to improve accuracy. The BERT model, with its remarkable ability to understand contextual nuances, has demonstrated high accuracy and efficiency, indicating its suitability for handling the diverse language and content present on social media.

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

The existing system for addressing cyber bullying typically relies on manual monitoring, user reporting, and intervention by platform administrators. Social media platforms may employ keyword filtering and rule-based algorithms to flag potential instances. However, these approaches often lack precision and may result in false positives or miss subtle forms of cyber bullying. Automated detection systems, particularly machine learning models, are increasingly being explored to enhance accuracy and efficiency in identifying and combating cyber bullying incidents.

3.1.1Dataset

The **Cyber Saver Detection of Cyber Bullying** dataset is designed to help identify and classify instances of cyber bullying in online interactions, particularly on social media platforms or messaging systems. It typically consists of textual data, such as comments, messages, or posts, labeled as either bullying or non-bullying. The dataset includes features like the content of the text, user information, and metadata that can be analyzed using machine learning models. The goal is to develop algorithms that can automatically detect harmful or abusive language and flag it for moderation. This type of dataset is essential for building systems that can help mitigate the negative impacts of cyber bullying by identifying and addressing harmful content in real-time.

3.1.2 Data pre processing

Pre-processing the **Cyber Saver Detection of Cyber Bullying** dataset involves several steps to clean and prepare the data for analysis and model training. First, raw text data is cleaned by removing irrelevant elements such as URLs, special characters, punctuation, and extra whitespace. Then, the text is normalized by converting all characters to lowercase and correcting any inconsistencies in spelling. Tokenization is performed to split the text into individual words or phrases, followed by removing, are then used to convert the text data into numerical representations suitable for machine learning models. Finally, the dataset is split into training and testing sets, and the labels are encoded for classification. These pre-processing steps help ensure that the data is in an optimal format for detecting and classifying instances of cyber bullying.

3.1.3 Model evolution

The Naive Bayes algorithm is a powerful and straightforward machine learning method used in the Cyber Saver - A Machine Learning Approach to Detection of Cyber Bullying system. In this context, Naive Bayes is employed to classify text data as either indicative of cyber bullying or not. The process begins by training the model on a labelled dataset of online messages, where each message is tagged as either "cyber bullying" or "non-cyber bullying."

Naive Bayes operates under the assumption of conditional independence between words given the class, which simplifies the model's training process. It calculates the probability of a message belonging to each class based on the frequencies of words appearing in the message. Using Bayes' Theorem, the algorithm combines these word probabilities to predict the most likely class for any given input message. After training, the model is capable of analysing new, unseen text and classifying it as either bullying or non-bullying based on the learned probabilities. Despite its simplicity, Naive Bayes works well in text classification tasks like cyber bullying detection due to its efficiency and ability to handle large datasets. However, its performance may vary depending on the quality of the feature extraction process

DISADVANTAGES

1. Manual monitoring lacks efficiency.
2. Rule-based filters may produce false positives.
3. Limited adaptability to evolving cyber bullying tactics

3.2 PROPOSED SYSTEM

The proposed system utilizes machine learning algorithms, including linear Support Vector Classification (SVC), for precise detection of cyber bullying in text and images. This proactive approach enhances accuracy and responsiveness, providing a more efficient and automated means to address cyber bullying incidents.

3.2.1 Dataset

The **Cyber Saver Detection of Cyber Bullying** dataset is designed to help identify and classify instances of cyber bullying in online interactions, particularly on social media platforms or messaging systems. It typically consists of textual data, such as comments, messages, or posts, labeled as either bullying or non-bullying. The dataset includes features like the content of the text, user information, and metadata that can be analyzed using machine learning models.

The goal is to develop algorithms that can automatically detect harmful or abusive language and flag it for moderation. This type of dataset is essential for building systems that can help mitigate the negative impacts of cyber bullying by identifying and addressing harmful content in real-time.

3.2.2 Data pre processing

Pre-processing the **Cyber Saver Detection of Cyber Bullying** dataset involves several steps to clean and prepare the data for analysis and model training. First, raw text data is cleaned by removing irrelevant elements such as URLs, special characters, punctuation, and extra whitespace. Then, the text is normalized by converting all characters to lowercase and correcting any inconsistencies in spelling. Tokenization is performed to split the text into individual words or phrases, followed by removing, are then used to convert the text data into numerical representations suitable for machine learning models. Finally, the dataset is split into training and testing sets, and the labels are encoded for classification. These pre-processing steps help ensure that the data is in an optimal format for detecting and classifying instances of cyber bullying.

pre-process the data, train a Support Vector Machine (SVM) classifier on this data, and then use the trained model to predict whether a new loan applicant is likely to be approved or not; essentially classifying them based on their features compared to historical data SVM chooses the extreme points/vectors that help in creating the hyper plane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyper plane:

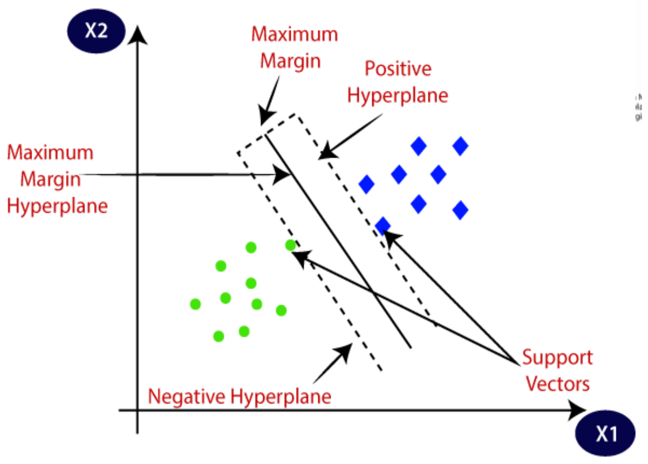


Fig 3.1 Analysis of SVM

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

SVM working

The Linear Support Vector Machine (SVM) algorithm aims to find the optimal decision boundary, or hyperplane, that separates different classes in the dataset. In the case of a 2-dimensional space, this boundary is simply a straight line. However, as shown in the example, there can be multiple lines that can separate the two classes, making it important to choose the line that best generalizes the data. This optimal line is the one that maximizes the margin between the two classes, which is defined as the distance between the closest points of each class to the line. These closest points are known as support vectors, and they are critical in determining the position of the hyperplane.

In the context of SVM, the goal is to maximize this margin to ensure that the classifier performs well on unseen data. The larger the margin, the better the SVM will generalize to new, previously unseen data points. While there can be multiple lines that separate the classes, the one with the maximum margin is the optimal choice. This principle is known as the Maximum Margin Classifier. The SVM algorithm achieves this by solving a quadratic optimization problem that finds the hyperplane that maximizes the margin while ensuring that all data points are classified correctly. This approach makes SVM particularly effective for binary classification tasks.

Linear SVM: The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair (x1, x2) of coordinates in either green or blue. Consider the below image:

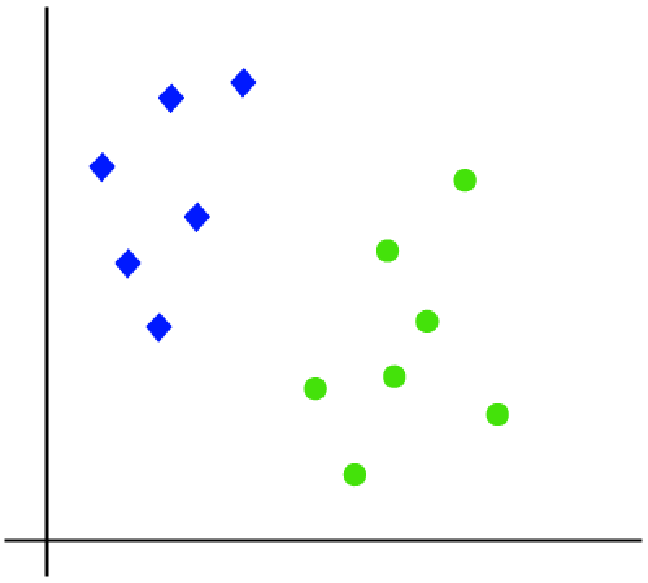


Fig 3.2 Linear SVM

So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

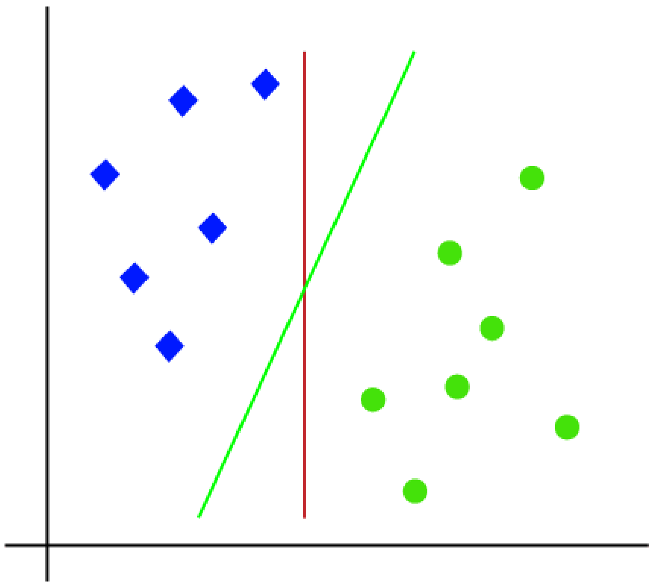


Fig3.3 Test-Vector in SVM

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyper plane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyper plane is called as margin. And the goal of SVM is to maximize this margin. The hyper plane with maximum margin is called the optimal hyper plane. By applying this model to new loan applications, it can predict the likelihood of cyber bullies, helping financial institutions make informed decisions. The SVM model is particularly effective in handling high-dimensional data and can be tuned for better accuracy using techniques like kernel trick, cross-validation, and hyper parameter optimization. In conclusion, Linear Support Vector Machine (SVM) is a powerful and effective algorithm for binary classification tasks, particularly when the data is linearly separable. By finding the optimal hyperplane that maximizes the margin between different classes, SVM ensures robust generalization to unseen data, minimizing the risk of overfitting. While the algorithm excels in linearly separable cases, it also provides flexibility through the introduction of the soft margin, allowing for some misclassifications to improve model performance on noisy or imperfect data. Furthermore, SVM’s capability to handle non-linearly separable data through the kernel trick extends its utility to complex classification problems. Overall, SVM remains a popular and reliable choice for a wide range of applications, offering a strong balance of accuracy, efficiency, and adaptability in various domains.

ADVANTAGES

1. Enhanced precision with linear SVC.
2. Automated detection improves efficiency.
3. Proactive approach addresses evolving cyber bullying threats with machine learning algorithms.

CHAPTER-4

SYSTEM STUDY

4.1 FEASIBILITY STUDY

Checking the project's viability and submitting a business proposal outlining the project's broad objectives, scope, and anticipated expenses constitute this step. Finding out if the proposed model is workable is the main goal of system analysis. It is crucial to guarantee that the recommended cure will not put the organization in a financial bind. Finding out what your system really needs is the first thing you should do when undertaking a practical analysis.

There are three main aspects that must be considered in order to conduct a feasibility study successfully:

Infrastructure Requirements: You need to check the present telecom network to see whether it can handle the extra computing demand. Data storage options, server capacity, and network bandwidth are all part of this.

Integration: Find out how simple it is to incorporate the recommender systems into preexisting databases, call logs, & telecommunications software. Importantly, it must be compatible with preexisting data gathering and processing infrastructure.

Scalability: Make sure the solution can scale to accommodate more users and more call data without sacrificing speed.

4.2 ECONOMICAL FLEASIBILITY:

The primary objective of this analysis is to determine the system's actual earnings for the company. The amount of money the corporation can put into the early phases of developing the system is limited. It is necessary to reassess the budget. The majority of the used technologies are open source and may be utilized for free, which is the main rationale for this. The one and only requirement was to buy the personalized products.

4.3 TECHNICAL FEASIBILITY:

The results of this study validate the system's technical feasibility. It is best to build systems in a way that doesn't strain current infrastructure too much. Because of this, our limited technological resources will definitely be challenged. This could lead to a great deal of pressure on buyer. There shouldn't be many needs for the created system since there won't be many adjustments to implement it.

4.4 SOCIAL FEASIBILITY:

Research is being conducted to determine the extent to which people trust the system. Part of the process is making sure the user knows how to get the most out of the technology. Making sure the user doesn't feel intimidated or afraid while using the system is crucial. An individual's investment in the system is directly correlated to the effort they put into familiarizing themselves with its features and workings. It is essential that he has the self-assurance to provide the much-needed positive feedback as he is system's end user.

Legal and Ethical Feasibility

Compliance: Make that the system follows all local and national legislation regarding data protection & telecommunications, including GDPR.

Customer Privacy: Take strong precautions to safeguard customer data & preserve privacy. Be clear about how your company uses customer data.

Technical Risks: Problems with data precision, model performance, or system integration are examples of technical risks.

Operational Risks: Difficulties with Data Gathering, System Upkeep, and User Acceptance.

Economic Risks: Spending more than planned and getting less return on investment than intended.

Legal Risks: Possible data breaches & failure to comply with rules.

Ensuring compliance with relevant laws and regulations is vital for the system's success. It must adhere to data protection laws like the General Data Protection Regulation (GDPR), ensuring that personal data is securely handled. This includes obtaining consent from users, clearly communicating how their data will be used, and allowing them to withdraw consent if desired. Safeguarding sensitive customer information with encryption and access controls is essential to maintaining privacy.

Ethically, the system must inform users about how their data will be used and ensure their participation is voluntary. Clear communication about privacy protections can help build trust. Additionally, addressing any biases in the model to prevent unfair treatment is important. Ethical considerations should ensure that the system doesn’t harm users or violate their rights.

CHAPTER-5
SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE:

The **Cyber Saver Detection of Cyber Bullying** system architecture typically follows a multi-layered approach that includes data collection, preprocessing, model training, and deployment. The architecture starts with **data acquisition**, where raw text data from social media platforms or messaging systems is gathered. Next, **data preprocessing** is performed to clean, normalize, and transform the text into a format suitable for analysis, often involving techniques like tokenization, stop word removal, and stemming. The processed data is then fed into **a machine learning model**

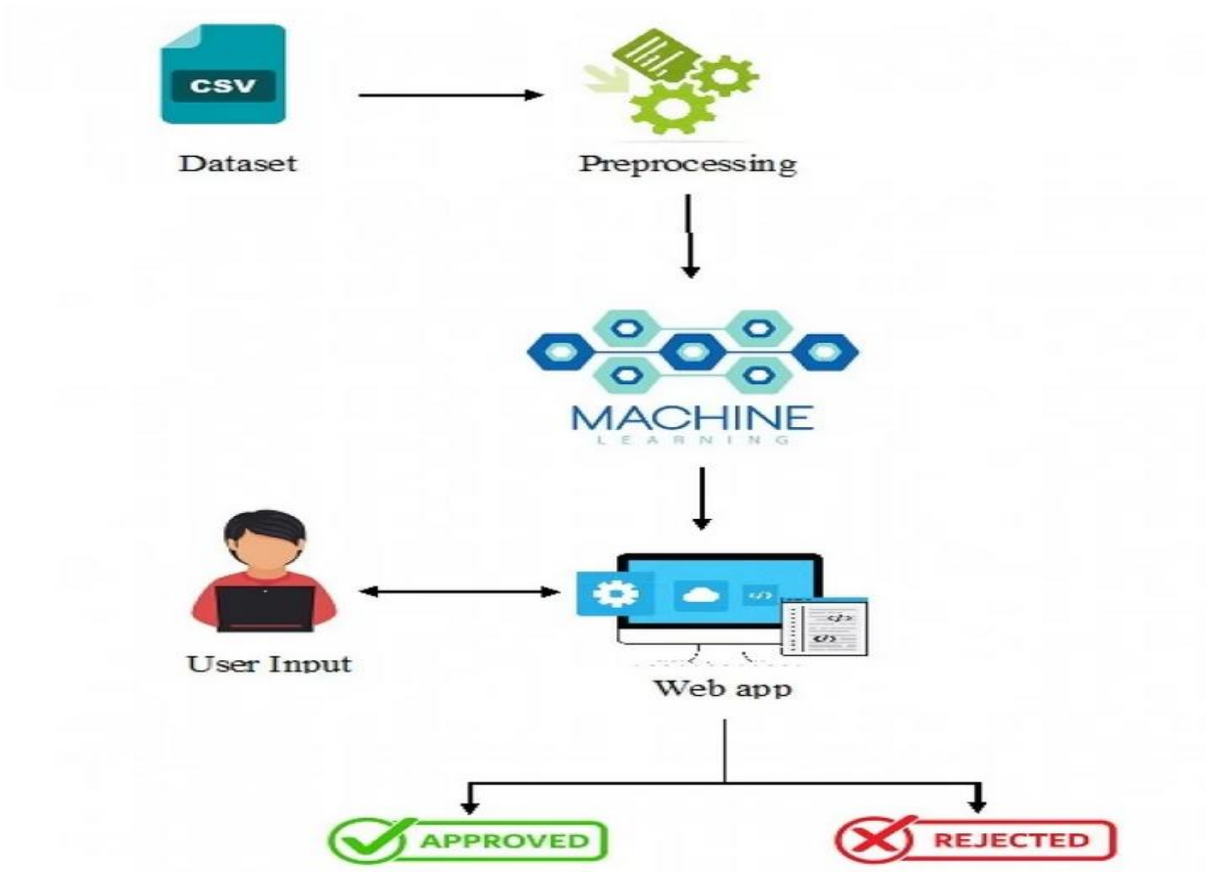


Fig 5.1 Steps of Process

The Cyber Saver Detection of Cyber Bullying system architecture is designed to efficiently detect and prevent cyberbullying by following a structured workflow. It begins with **data collection**, where raw data, typically consisting of social media posts, comments, or messages, is retrieved from platforms like Twitter, Facebook, or online forums. This data is then subjected to **data preprocessing**, which involves several steps such as **tokenization** (breaking the text into smaller units like words), **stop-word removal** (eliminating common but unimportant words), and **stemming** (reducing words to their root form). The preprocessed data is subsequently fed into a **machine learning model**, such as a Support Vector Machine (SVM), BERT, or deep neural networks, which are trained to identify patterns indicative of cyberbullying. The model is trained using labeled datasets, where examples of cyberbullying and non-cyberbullying content are provided. Once trained, the system can classify new, unseen data in real-time, providing alerts or reports on potential cyberbullying instances. Finally, the system is deployed in a live environment, where it continuously monitors incoming messages, provides detection results, and can trigger actions such as user warnings or reporting mechanisms. This multi-layered approach ensures that the system is robust, scalable, and capable of adapting to the dynamic nature of online communication.

5.2 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

USER:

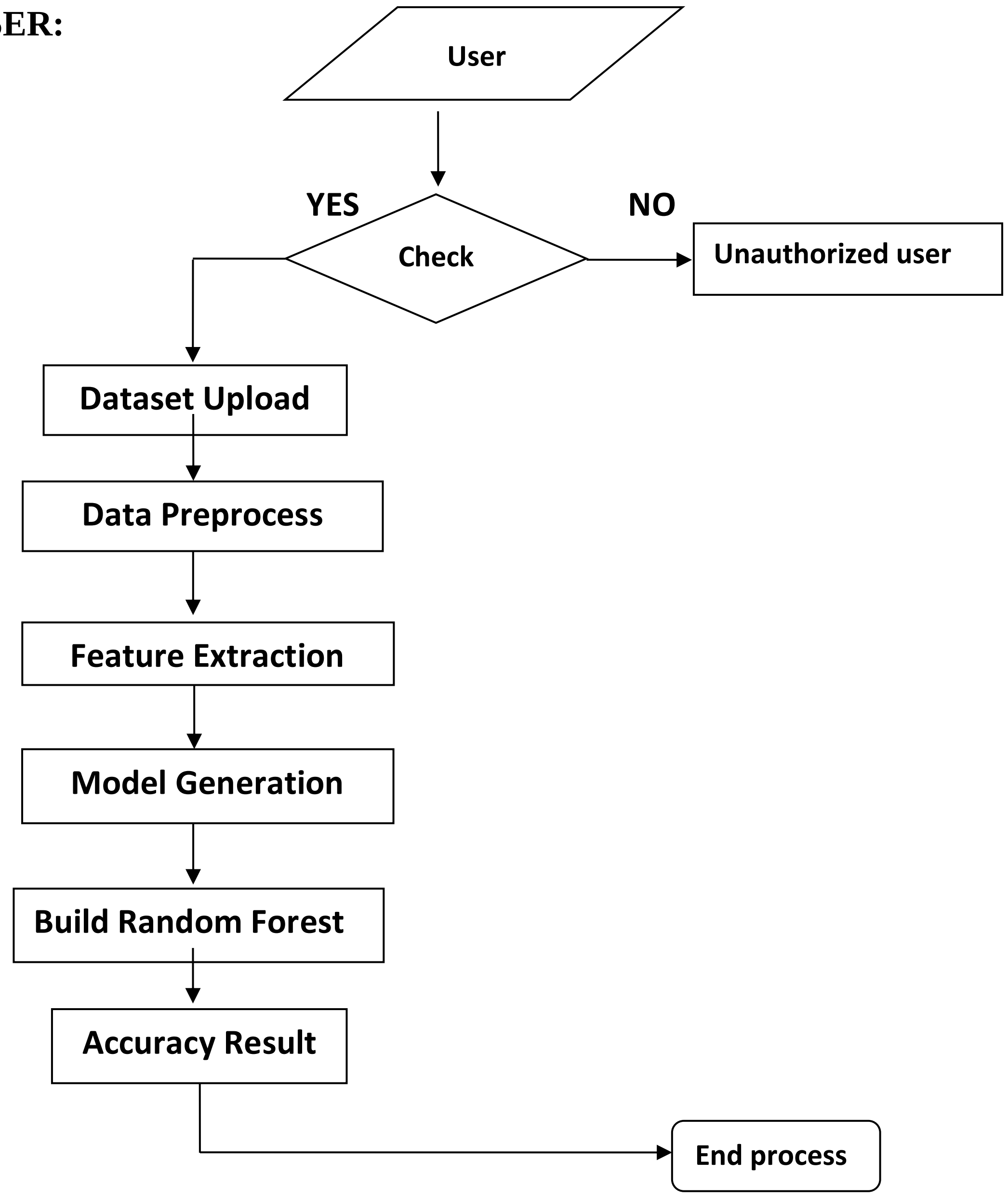


Fig 5.2 Dataflow Diagram

5.3 UML DIAGRAMS

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

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

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

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

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

GOALS:

1. The Primary goals in the design of the UML are as follows:
2. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
3. Provide extendibility and specialization mechanisms to extend the core concepts.
4. Be independent of particular programming languages and development process.
5. Provide a formal basis for understanding the modelling language.
6. Encourage the growth of OO tools market.
7. Support higher level development concepts such as collaborations, frameworks, patterns and components.
8. Integrate best practices.

5.3.1 Use case diagram:

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

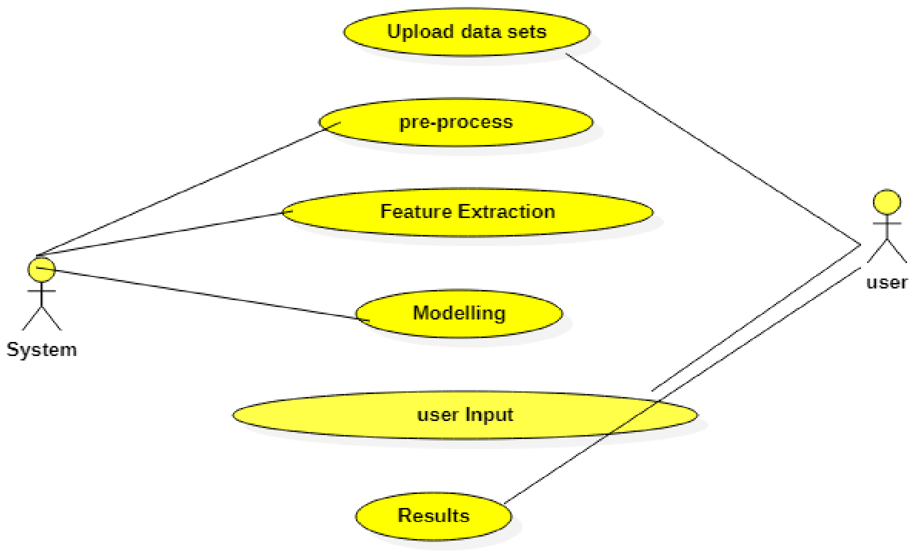


Fig 5.3 Use case Diagram

5.3.2 Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

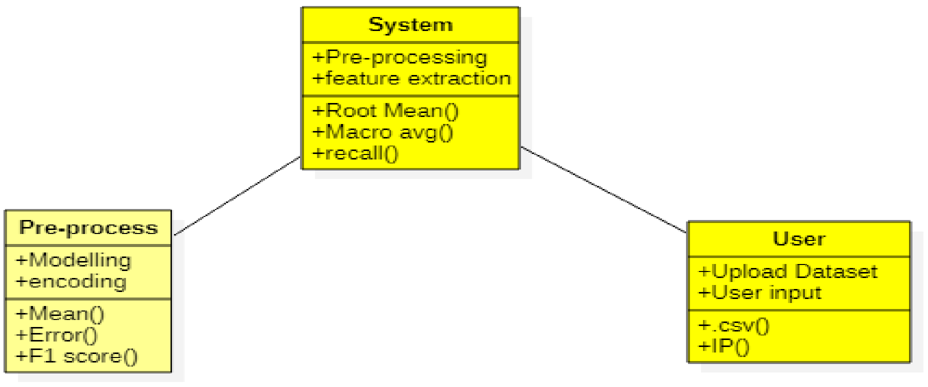


Fig 5.4 Class Diagram

5.3.3 Object diagram:

The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while the system is running. The object diagram captures the state of different classes in the system and their relationships or associations at a given point of time.

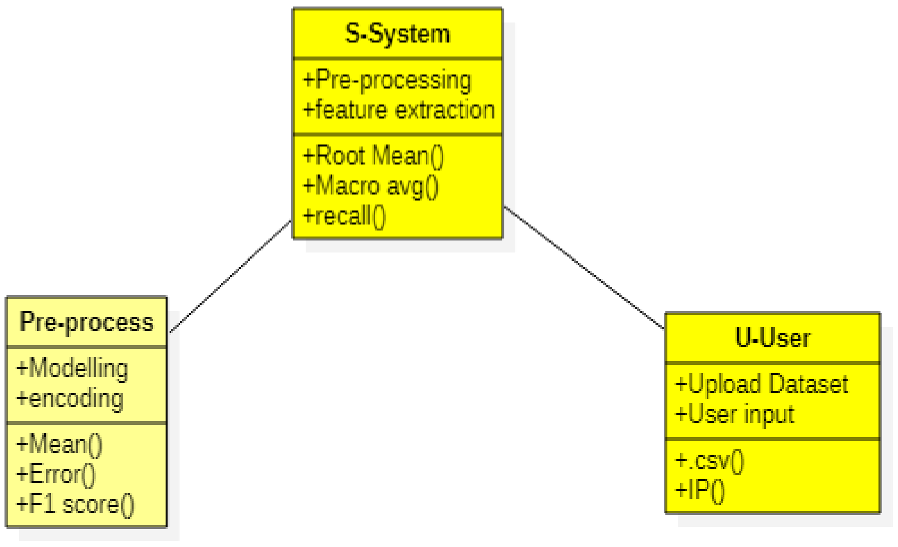


Fig 5.5 Object Diagram

5.3.4 State diagram:

A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.

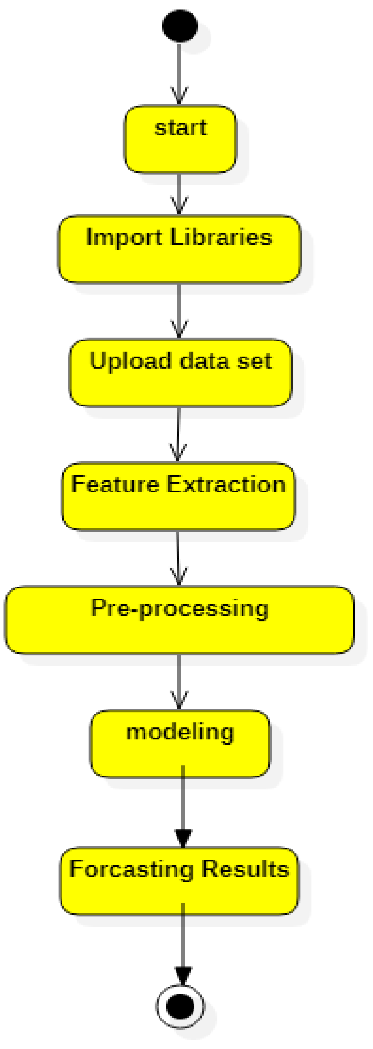


Fig 5.6 State Diagram

5.3.5 Activity diagram:

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

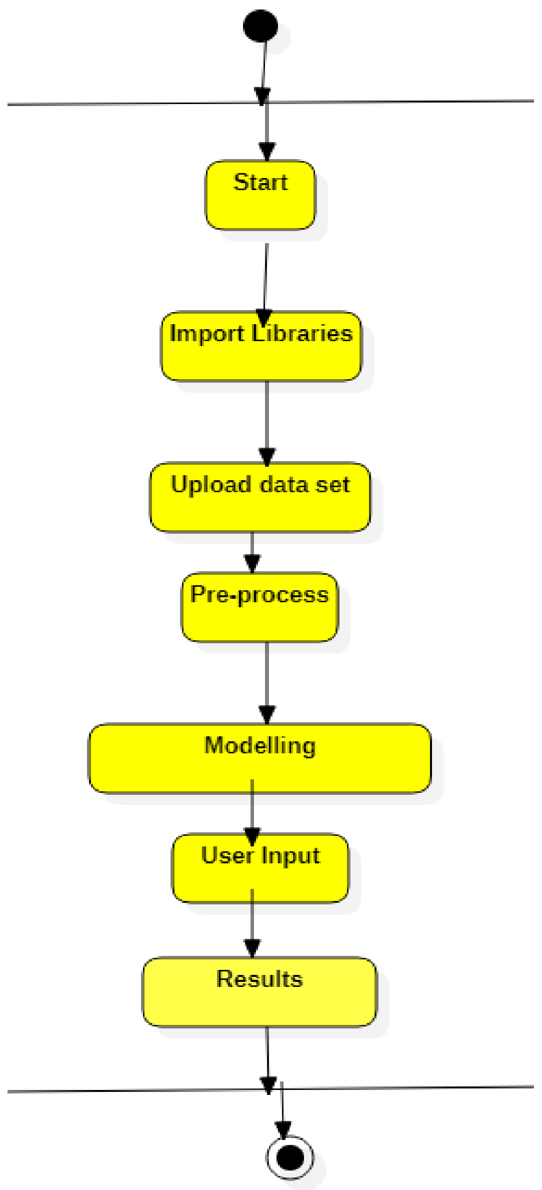


Fig 5.7 Activity diagram

5.3.6 Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

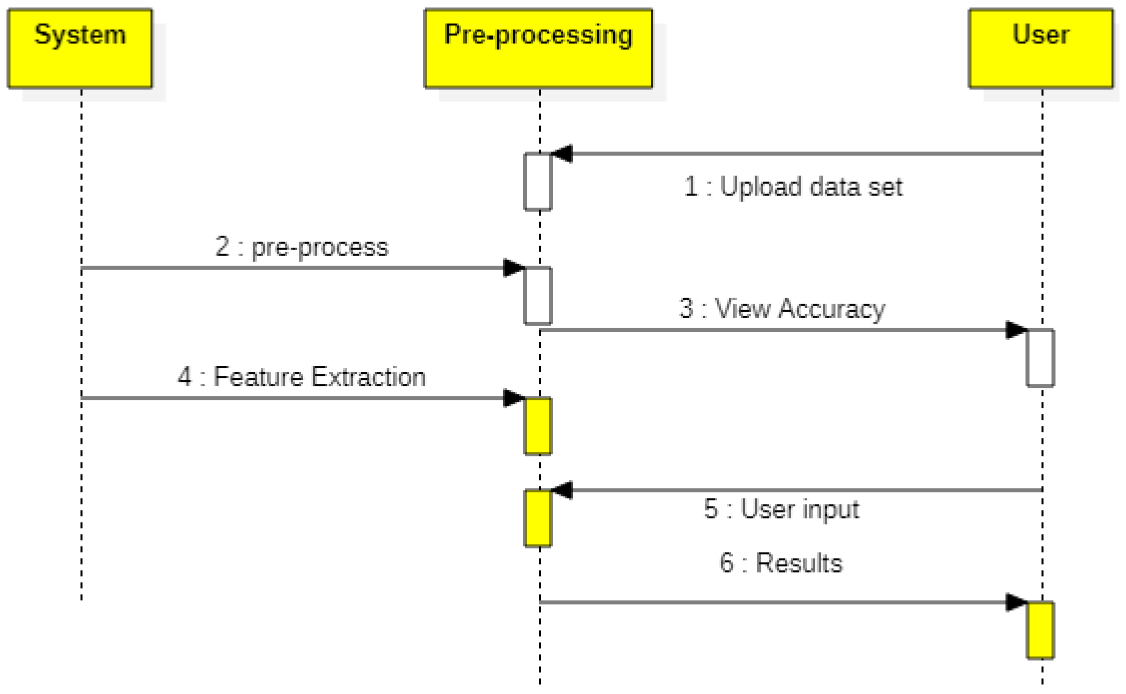


Fig 5.8 Sequence Diagram

A sequence diagram is a type of interaction diagram used in object-oriented software design to visually represent how objects or components of a system interact with each other over time. It emphasizes the order and flow of messages between these objects, ensuring that the interactions are presented in a clear, step-by-step, time-ordered sequence. Each object or participant in the system is typically represented by a vertical lifeline, with horizontal arrows representing the messages that are passed between these objects. The sequence of events is depicted along the time axis, with the flow of messages occurring from top to bottom, allowing developers and stakeholders to clearly see the order in which operations are performed.

.

CHAPTER-6

SYSTEM REQUIREMENTS

6.1 SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

- Python IDLE 3.7 or 3.10 version
- PYcharm
- Jupiter (or) HTML or CSS
- Django or Flask or Starlit or tinkter

6.2HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

• Operating system	:	Windows, Linux
• Processor	:	minimum intel i3
• Ram	:	minimum 4 GB
• Hard disk	:	minimum 250GB

6.3 FUNCTIONAL REQUIREMENTS

Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

Output Definition

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

Input Design

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

Input Stages

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

Input Types

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

Input Media

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

Error Avoidance

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

Error Detection

Even though every effort is to make to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

Data Validation

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

User Interface Design

It is essential to consult the system users and discuss their needs while designing the user interface:

User Interface Systems Can Be Broadly Classified As:

- User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
- Computer initiated interfaces in the computer-initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

User Initiated Interfaces

User initiated interfaces fall into two approximate classes:

- Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
- Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

Computer-Initiated Interfaces

The following computer – initiated interfaces were used:

- The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
- Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

Error Message Design

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed. This application must be able to produce output at different modules for different inputs.

Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment.

It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system
- The existing system is completely dependent on the user to perform all the duties.

CHAPTER-7
SOFTWARE ENVIRONMENT

7.1 INTRODUCTION TO MACHINE LEARNING:

The term "Machine Learning" was first used by Arthur Samuel in 1959 to describe the "Field of study that gives computers the capability to learn without being explicitly programmed." The word spread. As a result, the study of machine learning came into being. Machine learning is one of the most sought-after careers in the modern economy. Among all the jobs in 2019, Machine Learning Engineer has the highest average base salary at $146,085 and the fastest job growth rate at 344%, according to Indeed. Still, a lot of people have no idea what Machine Learning is or how it works. Become an expert in machine learning by reading this essay, which covers the principles of the topic.

Alright, then, I will start! The intricacies of various machine learning methodologies may be explored later; for now, let's take a look at what machine learning is not. Although many people see machine learning as a subfield of artificial intelligence, I disagree. Machine learning certainly has its roots in this area, but it really shines when seen through the lens of data science as a tool for build data models. The fundamental idea behind machine learning is that mathematical models may help us understand data better.

We introduce the idea of "learning" when we provide these models tweakable parameters that may be adjusted depending on observed data; in this way, the computer is "learning" from the data. After being fitted to historical data, these models may be used to understand and forecast aspects of newly encountered data. The degree to which this "learning" based on mathematical models matches the "learning" that the human brain accomplishes is a philosophical aside, and I'll leave it up to the reader to determine. Having a firm grasp of machine learning's issue setting is crucial for making effective use of these techniques. In light of this, we will start by offering some broad categories for the methods that will be covered.
Machine learning is a technique that examines datasets in search of certain patterns.
1.It has the potential to improve its performance on its own by looking at past data.
2.The technique is based on data there is a lot of overlap between data mining and machine learning as both deal with large datasets.

7.2 How does Machine Learning Work?

Machine learning systems may build prediction models using historical data, apply those models to new data, and finally, provide accurate forecasts. More data means a more robust model, which means more accurate forecasts, hence data quantity is king when it comes to predicting output accuracy. Machine learning tasks may be broadly classified into many categories. In a supervised learning approach, a mathematical model is built using both the inputs and the outputs of a data collection.

The input data would include both photographs with and without the object, & output data will indicate whether each image included the object, if the objective was to train a supervised learning algorithm to recognize the presence or absence of a given item in photos. On occasion, the input is partially available or restricted to certain types of feedback. It is possible for semi-supervised learning algorithms to construct mathematical models using input samples that do not all contain labels.

Several approaches, including as regression and classification algorithms, are part of supervised learning. Classification algorithms are useful when there are few alternative values for the outputs. One way to classify incoming emails is as follows: the method takes the email as input and returns the name of the folder where the email should be filed. The prediction of "spam" / "not spam," represented by the Boolean values true and false, respectively, might be one potential output of an algorithm for spam email detection.

Regression methods provide continuous outputs, which may take on any value within a specified range. Things that may be expressed as a continuous value include length, temperature, & price. 17 Algorithms may build mathematical models in unsupervised learning without labels for the model's outputs. Using unsupervised learning approaches, patterns in data may be discovered by grouping or clustering of data points. In addition to input classification, unsupervised learning, like feature learning, may discover data patterns.

Dimensionality reduction is the process of reducing the amount of features, or inputs, to a data collection. Active learning algorithms aim to acquire training labels by selecting the most appropriate inputs from a limited set of options within a predetermined budget. It is possible to display an interactive user objects and ask them to identify them. Reinforcement learning algorithms are used in both autonomous vehicles and learning to play games against human opponents.

These algorithms are educated in a dynamic environment utilizing both positive and negative reinforcement. One of the many specialized methods in machine learning is topic modelling, which entails providing a program with a set of natural language processing texts and instructing it to find other documents that cover similar topics.

Machine learning techniques may be used to find the unobservable probability density function in density estimation scenarios. Even algorithms with built-in error correction capabilities may acquire their own inductive bias. Developmental robotics involves teaching robots new skills in small pieces via self-directed learning, which involves interacting with humans and figuring out how to get about. These autonomous robots are guided by a variety of mechanisms, including imitation, motor synergy, active learning, and maturation.

7.3 Applications of Machines Learning:

Some have even gone so far as to call this the "golden year" of AI and ML, two of the most rapidly expanding areas of technology. It is useful for tackling a number of difficult, non-trivial real-world problems.

A few examples of ML in action are these: –

- Analyzing emotions
- Evaluating public opinion
- Preventing as well as detecting errors
- Predicting along with weather forecasting
- Analysis as well as forecasting of the stock market
- synthesized speech
- Recognition of voice
- Segmenting customers
- Identifying objects
- Prevention of fraud
- Preventing fraud
- Making product suggestions to customers when they purchase online

7.4 PYTHON:

Python, in its original 0.9.0 form, was published by Guido Van Rossum at outsources in February 1991. Previous versions of this library had support for handling exceptions, functions, and common data types such as list, dict, str, and more. Because of its module design, it was object oriented. Public release of Python 1.0 occurred in January 1994. The most notable additions to this version were the functional programming tools that Guido Van Rossum despised—map, reduce, lambda, and function. After a gap of six and a half years, Python 2.0 was unveiled in October 2000. This version included list comprehensions, a full garbage collector, and support for Unicode. Python 2.x versions continued to be popular for eight more years until Python 3.0 (also referred to as "Python 3000" or "Py3K") was released.

7.5 WHY THE NAME PYTHON?

No. The name wasn't bestowed by a terrifying snake. Rossum favored a comedy sitcom that aired in the late 1970s. "Python" was subsequently used after the name "Monty Python's Flying Circus" was first used History of Python Versions Declined for release in 1989 Notices issued by the government - 1990

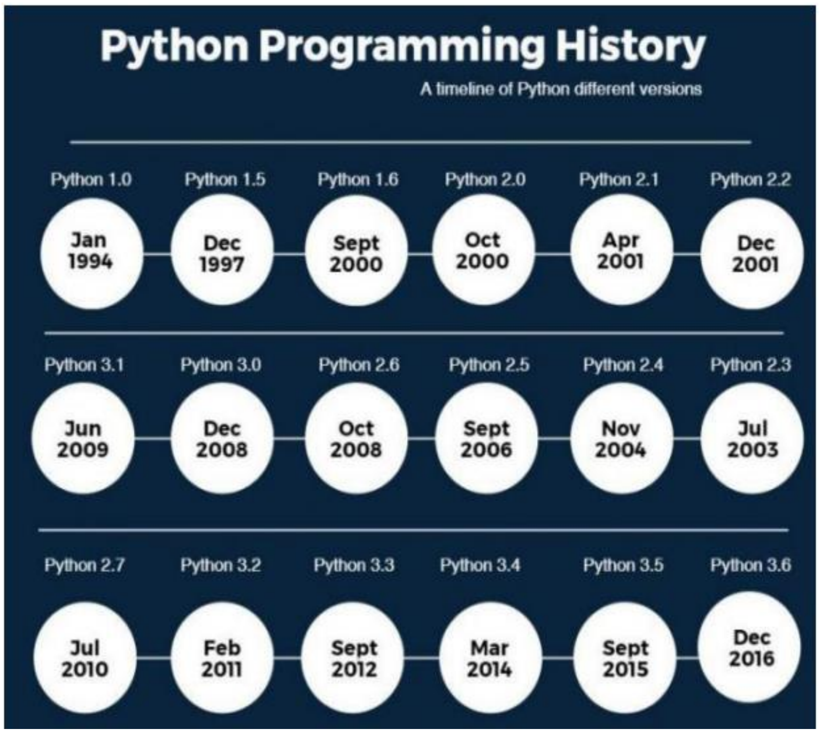


Fig 7.1 History of Python

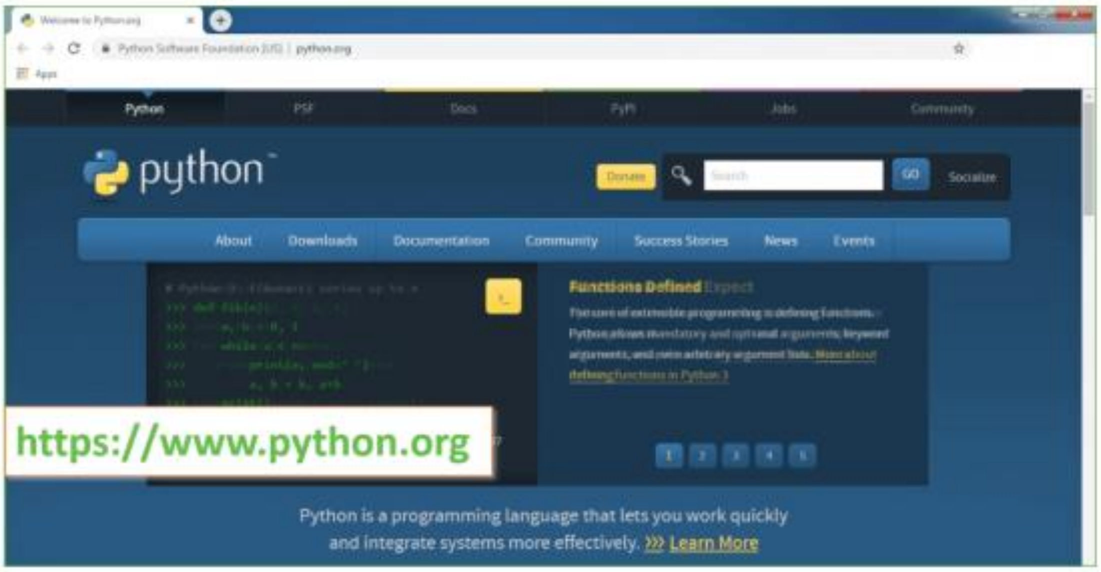
Install Python on Windows

- Getting it from the download page is the first step. Next, double-click the installation file to begin the setup procedure.
- Click Install Now after selecting Add Python 3.7 to PATH in setup window to begin installing Python 3.7

Install Python on Windows

- Getting it from the download page is the first step. Next, double-click the installation file to begin the setup procedure.
- Click Install Now after selecting Add Python 3.7 to PATH in setup window to begin installing Python 3.7

- Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

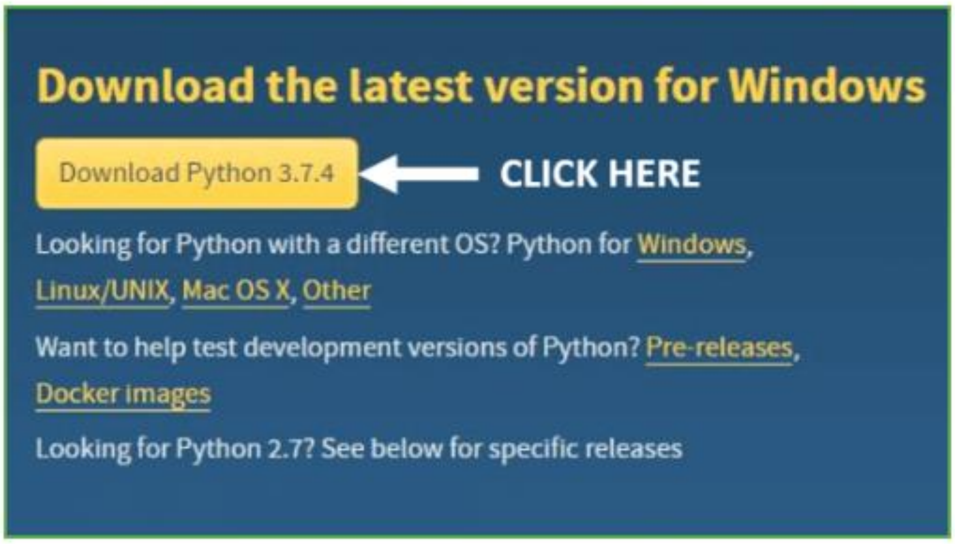


Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Fig 7.2 Python Installation

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Download the latest version for Windows

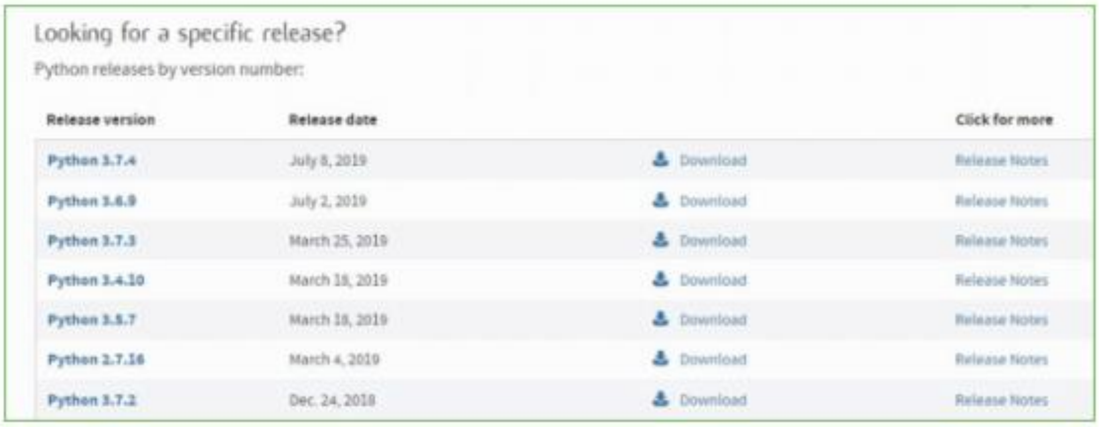
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Fig 7.3 Download the python 3.7.4

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Looking for a specific release?

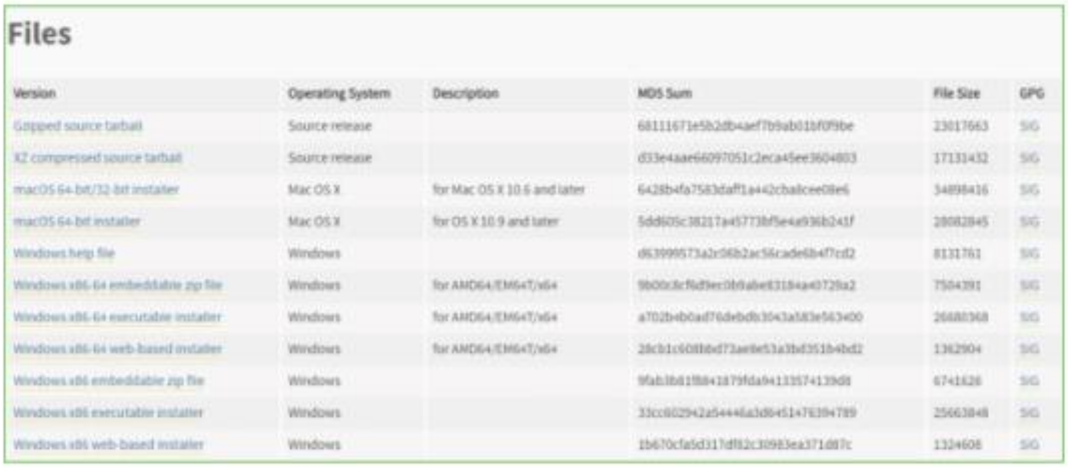
Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Fig 7.4 Versions of python

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c56b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Fig 7.5 Files of python

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

➢ Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

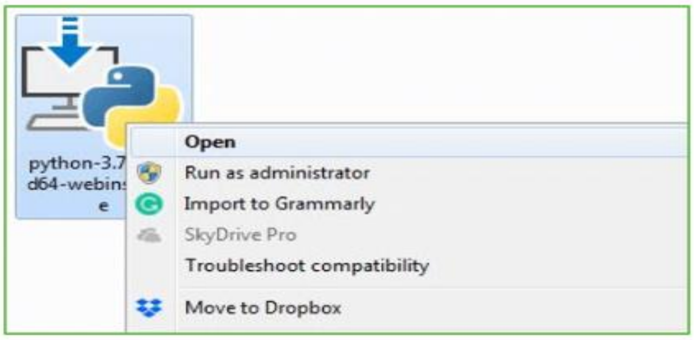
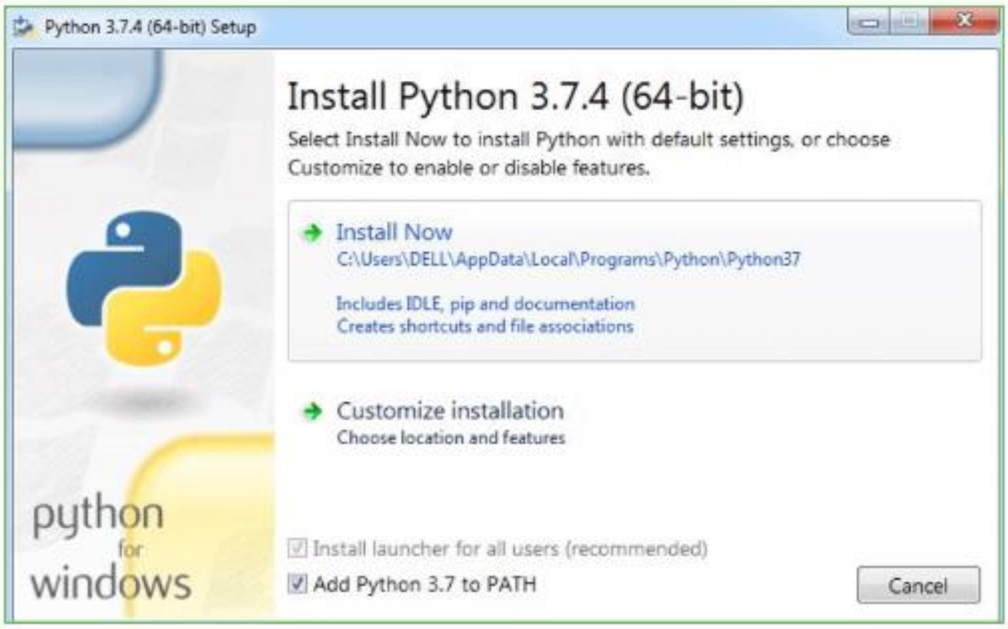


Fig 7.6 Opening the Application

Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.



Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

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

Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

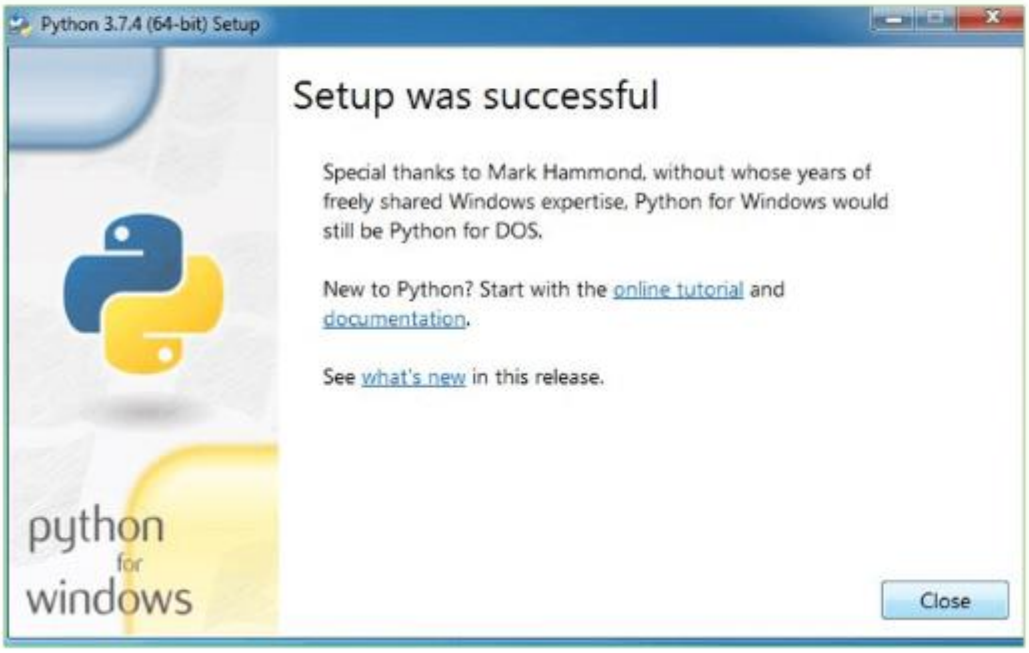
Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

[✓] Install launcher for all users (recommended)
[✓] Add Python 3.7 to PATH

Fig 7.7 Installation of Python

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Fig 7.8 Setup Process

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

- Verify the Python Installation

  Step 1: Click on Start

  Step 2: In the Windows Run Command, type “cmd”.

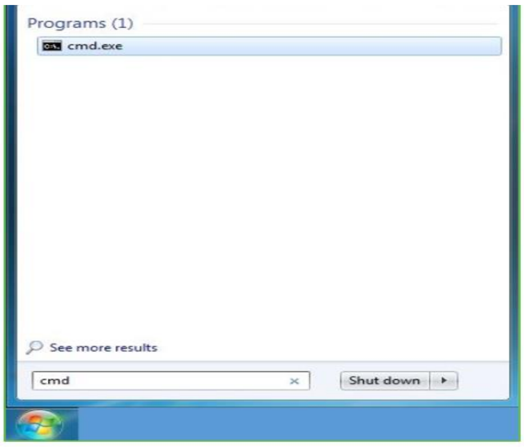
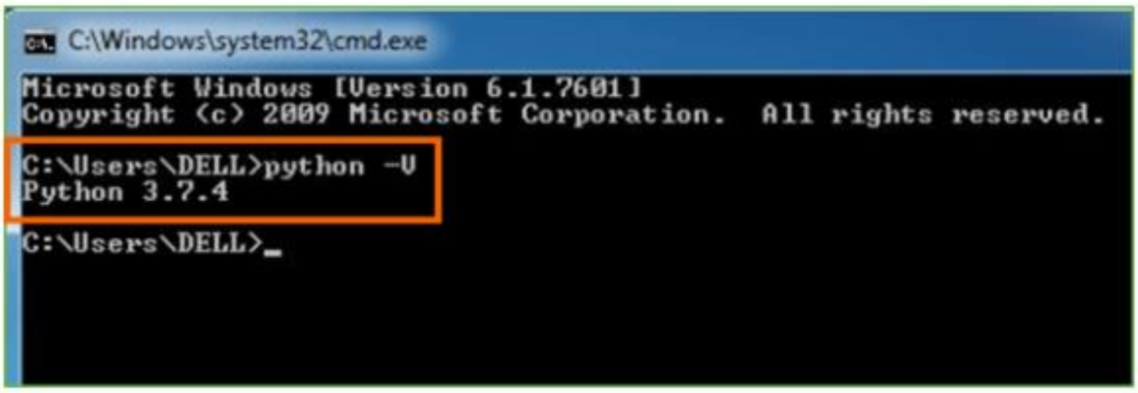


Fig 7.9 Opening by Command

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

Fig 7.10 checking the version of python in cmd

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.

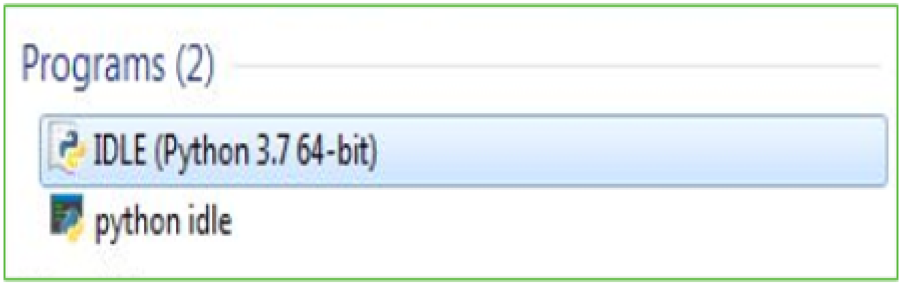
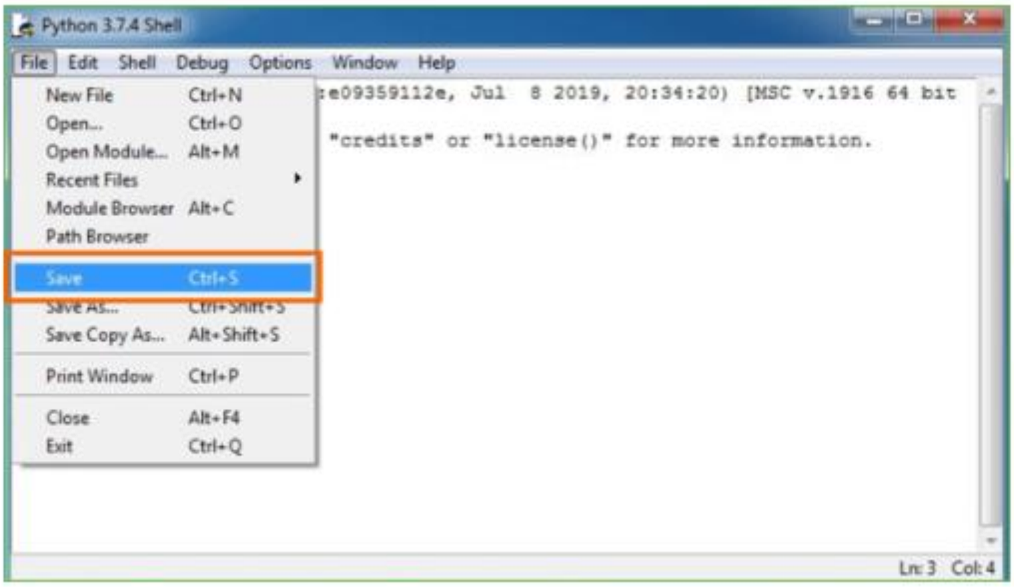


Fig 7.11 python idle 3.7 64-bit

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save



```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit
 "credits" or "license()" for more information.
```

Fig 7.12 opening python shell

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print

PACKAGES USED IN PROJECT:

Num Py

It is a general-purpose array processing tool. A high-performance object of dimensional arrays and tools to manipulate them are part of it. This is core module that any Python scientist needs to do their calculations.

It has several traits, the most important of which are:

A strong object that is an N-dimensional array.

Broadcasting operations that are complex

Tools for integrating C/C++ and Fortran applications

Skills in linear algebra, the Fourier transform, & useful operations on random numbers

Beyond its obvious scientific applications, Num Py's numerous uses lie in its efficiency as a multi-dimensional container of general data. Num Py's ease of use and support for a wide variety of data types makes it ideal for working with databases. One high-performance tool for data manipulation and analysis is Pandas, an open-source Python library known for its resilient data structures. Python was mostly used for data munging and preprocessing. In terms of data analysis, it was completely useless.

Pandas

No matter where the data comes from, Pandas makes it easy to conduct the five common processing and analysis steps: load, prepare, modify, model, and analyze. Python with Pandas is used by a wide variety of academic and professional fields, including economics, statistics, analytics, finance & economics.

Pandas is a Python library that provides easy-to-use and very effective data structures and analytics capabilities. Under the BSD approach, it is licensed and made available as open-source. Python with Pandas is used by numerous academic & professional fields, including economics, statistics, analytics, finance, and many more. In this session, we will delve into Python Pandas, a tool for data analysis.

A high-performance tool for data manipulation and analysis is provided by the open-source Python module Pandas, which uses its robust data structures. Pandas is derived from panel data, which is an econometric approach to multidimensional data. It was in 2008 that programmer Wes McKinney set out to build pandas, a flexible and fast tool for data analysis.

Prior to Pandas, data munging and preprocessing were Python's primary uses. In terms of data analysis, it was completely useless. Pandas fixed this problem. No matter where the data comes from, Pandas makes it easy to accomplish the five main steps of processing and analysis: loading, preparation, modification, modelling, and analysis. Numerous academic and professional fields use Python with Pandas, including economics, statistics, analytics, finance & economics.

Panda Characteristics:

This data frame object is both efficient and fast since it has both default & user-configurable indexing.

Tools for loading different kinds of files into data objects that are stored in memory Integrated coordination of data points and resolution of missing data

Tensor flow

Among free and open-source libraries, Tensor Flow stands out as the one to use for dataflow and differentiable programming. This symbolic math library is used by neural networks and other machine learning applications. It serves dual purposes in Google's research and production departments. Tensor Flow was developed by the Google Brain team specifically for use inside Google. It was made public on November 9, 2015, under the Apache 2.0 open-source license.

Matplotlib

Matplotlib is a Python 2D plotting toolbox that can produce figures which are fit for publishing, regardless of whether you're working in an interactive environment or with hardcopy forms. Matplotlib is compatible with a variety of Python modules and tools, such as scripts, the Python and I Python shells, the Jupiter Notebook, web application servers, & four GUI toolkits. Matplotlib aims to make complicated operations easier and simpler. Create a wide variety of graphs—from histograms & power spectra to bar charts, error plots, scatter plots, and more—with no scripting. For examples, have a look at the sample charts and the thumbnail galleries. When combined with I Python, the pyplot module's MATLAB-like interface makes it perfect for simple charting jobs. Power users have full control over line styles, font qualities, axis features, etc., using an object-oriented interface or a set of techniques common to MATLAB users.

Scikit – learn

The consistent-interface Python program Scikit-learn makes accessible a number of supervised and unsupervised learning methods. The software is released under a permissive simplified BSD license that promotes both academic and commercial use, and it is distributed under several Linux distributions.

Regression:

If the desired result incorporates one or more continuous variables, the procedure is referred to as regression. As an example of a regression issue, consider the problem of predicting the length of a salmon based on its age and weight. In unsupervised learning, the training data set consists of nothing more than a set of input vectors x, with no associated target values. Finding groups of similar data points may be accomplished in a number of ways; one of them is clustering; another is density estimation; & still another is data projection from a high-dimensional space to two or three dimensions for display.

The scikit-Learn package contains several popular datasets used in many fields. For example, the digits and iris datasets are used for classification, while the boston housing prices dataset is used in regression analysis.

Here, we bring in iris and numbers datasets after launching a Python interpreter from the command line. Typically, we'll be using the shell prompt ($) instead of Python interpreter prompt >>>:

```
$ python

>>> from sklearn import datasets

>>> iris = datasets.load iris()

>>> digits = datasets.load_digits()
```

A dataset contains all the data together with certain metadata. In many ways, it resembles a dictionary.

The data member contains an array called n_samples and n_features where this data is stored. In a supervised problem, target member keeps a response variable / variables. In the sections that follow, you'll discover more details on the different datasets. Take numbers as an example. The data gives you access to the features that may be used to classify samples of numbers in numbers dataset:

```
>>>print (digits. data)
```

```
[[ 0. 0. 5. ... 0. 0. 0.]

[ 0. 0. 0. ... 10. 0. 0.]

[ 0. 0. 0. ... 16. 9. 0.]

...

[ 0. 0. 1. ... 6. 0. 0.]

[ 0. 0. 2. ... 12. 0. 0.]

[ 0. 0. 10. ... 12. 1. 0.]]
```

The numerical value that corresponds to each digit image we are trying to learn is provided by Target for the digit dataset, which is the absolute truth.

SciKit-learn Genetic: This package uses evolutionary algorithms to optimize scikit-learn machine learning methods and perform feature selection. It is simple to import a scikit-learn regression / classification model, or a pipeline that incorporates one of these. We want to be able to provide a set of hyperparameters and range of feasible values for them, and this package aims to make that possible. The optimization process may be managed with the use of several specified optimization algorithms, call-backs, and built-in parameters. Starting with the most basic features and configurations, we will

dive right in evolutionary algorithms optimize using deep package. The technique works by first defining the set of hyper parameters to adjust from a randomly chosen population of possibilities. Then, it uses evolutionary operators for mating, mutation, selection, and evaluation to generate candidates with aim of raising the cross-validation score with each generation. The procedure will end when a certain amount of generations have elapsed or a specific callback condition has been met.

Keras:

When it comes to Deep Learning research and development, Tensor Flow and Theano are two of the top Python numerical frameworks. Both libraries are top-notch, but they might be a pain to work with when you're just starting out with deep learning model construction. If you're interested in learning how to construct different types of deep learning models using Tensor Flow or Theano, this article will introduce you to the Keras Python library. Keras is a basic Python package that can be used for deep learning and runs on top of Tensor Flow or Theano.

Its inception was motivated by a desire to alleviate the burden on the research and development community to quickly and easily access deep learning models. It runs well on CPUs and GPUs thanks to its base frameworks and needs either Python 2.7 or 3.5. Its distribution is governed by the relatively permissive MIT license. Four ideas were considered when Google engineer François Chollet developed and continues to develop Keras:

Modularity: The ability to comprehend a model via either a graph or a sequence is known as modularity. The various components of a deep learning model are loosely coupled and may be combined in any way the developer thinks appropriate.

Minimalism: The library presents just the information needed to complete a job, with a focus on simplicity and readability.

Extensibility: The purpose of facilitating the addition and use of additional components inside the framework is to promote experimentation & the testing new ideas by researchers. This is known as extensibility.

Python: No separate model files using proprietary file formats are available. Python is the foundation of our system. Having a Python and SciPy Environment ready to go makes installing Keras a breeze.

Installation of packages:

The syntax for installing packages in terminal using the standard

First step:1–Verify the pip command. If everything is OK,

Step 2: pip list.

To see what packages are already installed, use this command. You may then install the ones you need.

Step 3: install the package using pip.

When deciding on a package name, our needs should be considered.

CHAPTER-8

SYSTEM IMPLEMENTATION

8.1 MODULES

Dataset: Details about the dataset used for training and testing the machine learning model. This includes the size of the dataset, features, and labels.

Preprocessing: Steps taken to preprocess the data before feeding it into the machine learning model. This may involve tasks like data cleaning, handling missing values, and feature scaling.

Implementation Environment: Details about the programming language (e.g., Python), libraries, and frameworks used for implementing the machine learning model
project, particularly the linear Support Vector Classification (SVC). Include details about the parameters used for training the model.

Evaluation Metrics: Criteria used to evaluate the performance of the model. This could include metrics such as accuracy, precision, recall, F1 score, and confusion matrix.

8.2 SAMPLE CODE

```
import pandas as pd

import numpy as np

import re

import string

from nltk.stem import WordNetLemmatizer

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder

from nltk.tokenize import RegexpTokenizer

from nltk import PorterStemmer, WordNetLemmatizer
```

```
import pickle
import nltk
nltk.download('wordnet')
# preprocessing functions
# converting tweet text to lower case
def text_lower(text):
    return text.str.lower()
# removing stopwoords from the tweet text
def clean_stopwords(text):
    # stopwords list that needs to be excluded from the data
    stopwordlist = ['a', 'about', 'above', 'after', 'again', 'ain', 'all', 'am', 'an',
            'and', 'any', 'are', 'as', 'at', 'be', 'because', 'been', 'before',
            'being', 'below', 'between', 'both', 'by', 'can', 'd', 'did', 'do',
            'does', 'doing', 'down', 'during', 'each', 'few', 'for', 'from',
            'further', 'had', 'has', 'have', 'having', 'he', 'her', 'here',
            'hers', 'herself', 'him', 'himself', 'his', 'how', 'i', 'if', 'in',
            'into', 'is', 'it', 'its', 'itself', 'just', 'll', 'm', 'ma',
           'me', 'more', 'most', 'my', 'myself', 'now', 'o', 'of', 'on', 'once',
            'only', 'or', 'other', 'our', 'ours',' ourselves', 'out', 'own', 're',
            's', 'same', 'she', "shes", 'should', "shouldve",'so', 'some', 'such',
            't', 'than', 'that', "thatll", 'the', 'their', 'theirs', 'them',
            'themselves', 'then', 'there', 'these', 'they', 'this', 'those',
            'through', 'to', 'too', 'under', 'until', 'up', 've', 'very', 'was',
            'we', 'were', 'what', 'when', 'where', 'which', 'while', 'who', 'whom',
            'why', 'will', 'with', 'won', 'y', 'you', "youd","youll", "youre",
            "youve", 'your', 'yours', 'yourself', 'yourselves']
```

```
STOPWORDS = set(stopwordlist)
return " ".join([word for word in str(text).split() if word not in STOPWORDS])
# cleaning and removing punctuations
def clean_puctuations(text):
    english_puctuations = string.punctuation
    translator = str.maketrans('','', english_puctuations)
    return text.translate(translator)
# cleaning and removing repeating characters
def clean_repeating_characters(text):
    return re.sub(r'(.)1+', r'1', text)
# cleaning and removing URLs
def clean_URLs(text):
    return re.sub(r"((www.[^s]+)|(http\S+))","",text)
# cleaning and removing numeric data
def clean_numeric(text):
    return re.sub('[0-9]+', '', text)
# Tokenization of tweet text
def tokenize_tweet(text):
    tokenizer = RegexpTokenizer('\w+')
    text = text.apply(tokenizer.tokenize)
    return text
# stemming
def text_stemming(text):
    st = PorterStemmer()
    text = [st.stem(word) for word in text]
    return text
```

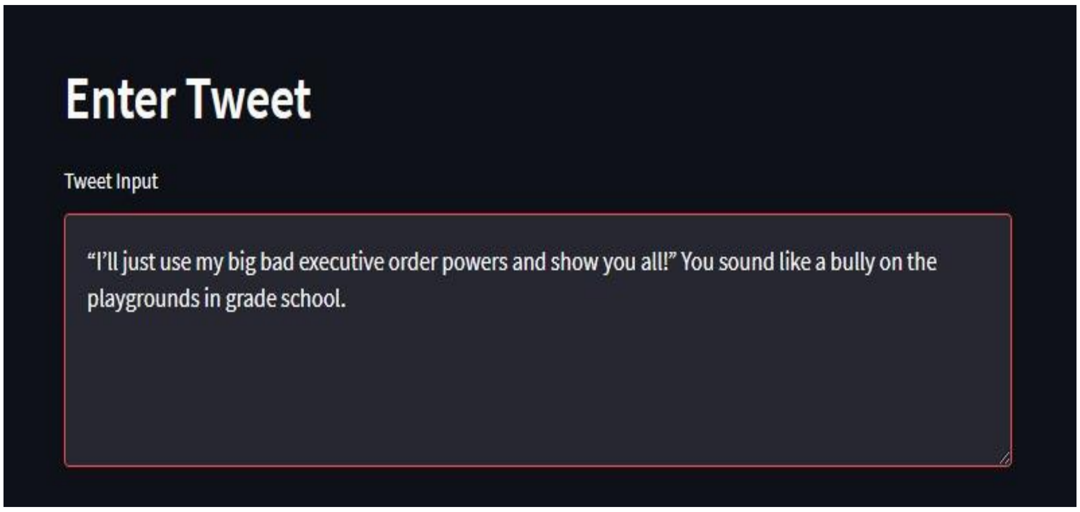
```
# lemmatization
def text_lemmatization(text):
    lm = WordNetLemmatizer()
    text = [lm.lemmatize(word) for word in text]
    return text
# defining preprocess function
def preprocess(text):
    text = text_lower(text)
    text = text.apply(lambda text: clean_stopwords(text))
    text = text.apply(lambda x : clean_puctuations(x))
    text = text.apply(lambda x: clean_repeating_characters(x))
    text = text.apply(lambda x : clean_URLs(x))
    text = text.apply(lambda x: clean_numeric(x))
    text = tokenize_tweet(text)
    text = text.apply(lambda x: text_stemming(x))
    text = text.apply(lambda x: text_lemmatization(x))
    text = text.apply(lambda x : " ".join(x))
    return text
# Function for custom input prediction
def custom_input_prediction(text):
    import nltk
    nltk.download('omw-1.4')
    text = pd.Series(text)
    text = preprocess(text)
    text = [text[0],]
```

```
# to use this function we will need to define vectoriser first
  vectoriser = pickle.load(open("tdf_vectorizer", "rb"))
  text = vectoriser.transform(text)
  model = pickle.load(open("model.bin", "rb"))
  prediction = model.predict(text)
  prediction = prediction[0]
  interpretations = {
      0: "Age",
      1: "Ethnicity",
      2: "Gender",
      3: "Not Cyberbullying",
      4: "Other Cyberbullying",
      5: "Religion"
  }

  for i in interpretations.keys():
      if i == prediction:
          return interpretations[i]
```

CHAPTER-9

RESULT


Enter Tweet

Tweet Input

“I'll just use my big bad executive order powers and show you all!” You sound like a bully on the playgrounds in grade school.

Fig 9.1 Age Bullying Tweet Input

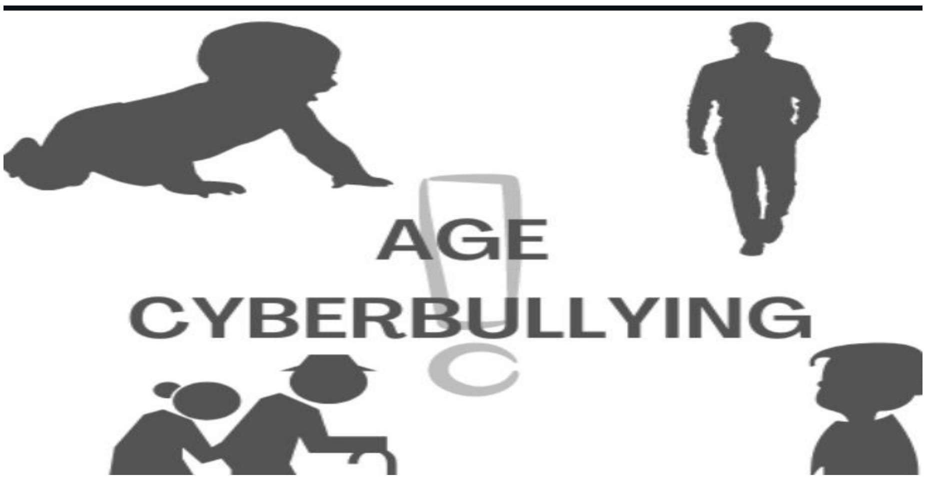
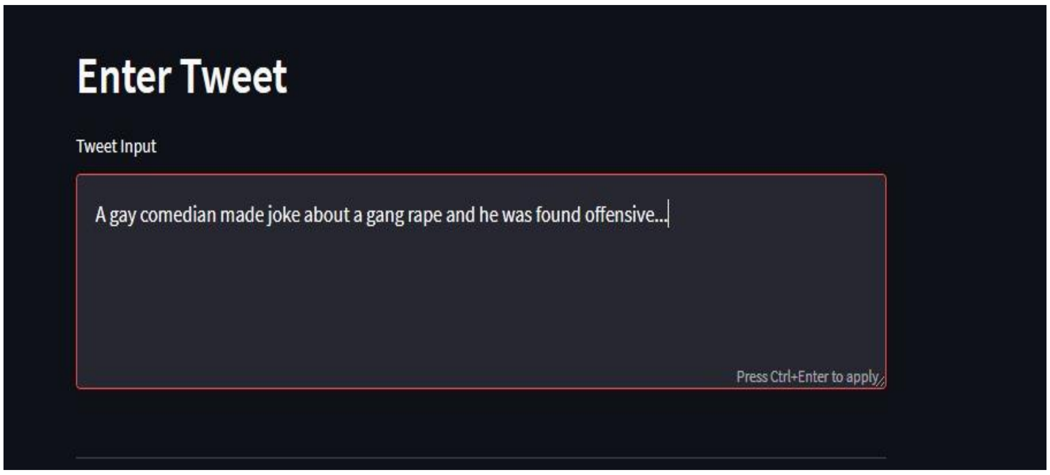


Fig 9.2 Age Bullying Output



Enter Tweet

Tweet Input

A gay comedian made joke about a gang rape and he was found offensive...

Fig 9.3 Gender Bullying Tweet Input

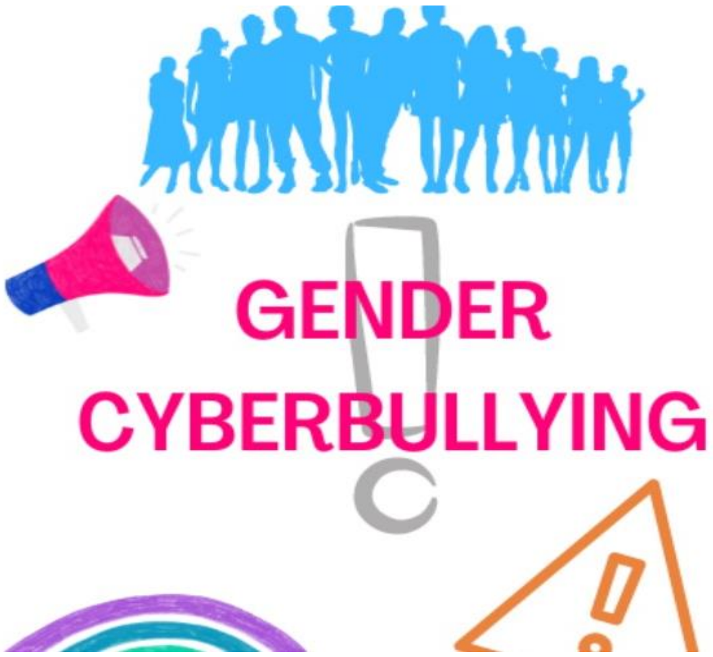
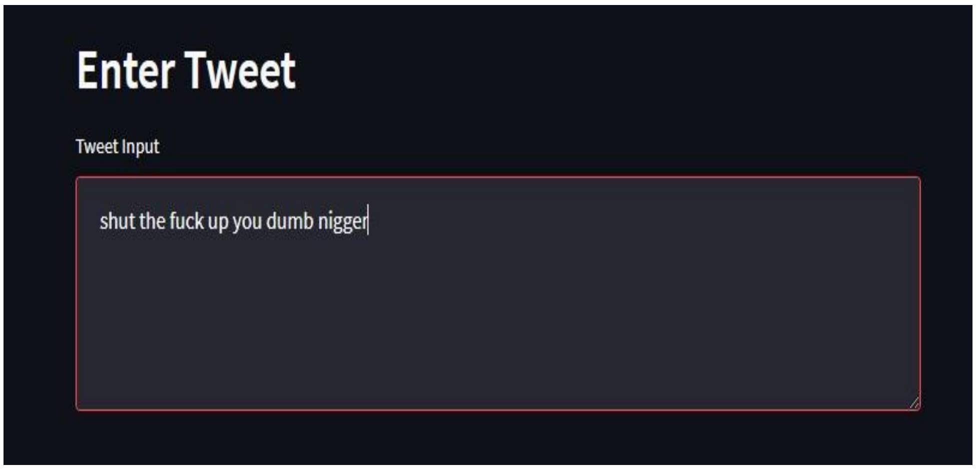


Fig 9.4 gender Bullying output



Enter Tweet

Tweet Input

shut the fuck up you dumb nigger

Fig 9.5 Ethnicity Cyberbullying Tweet Input

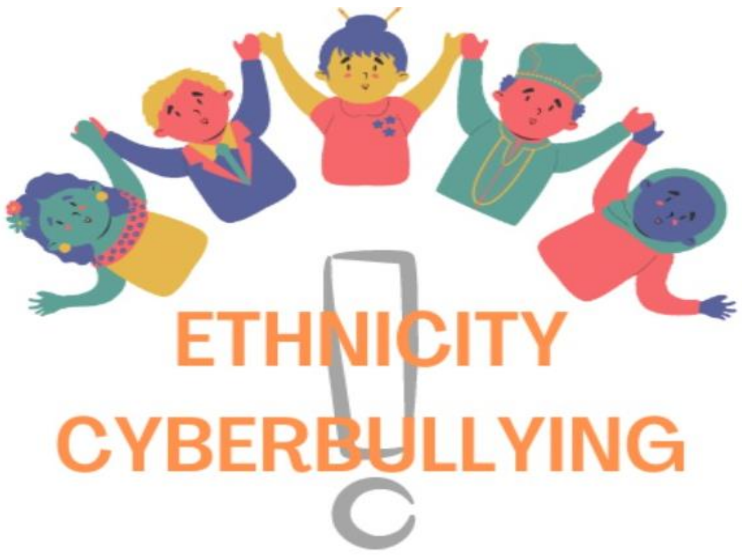
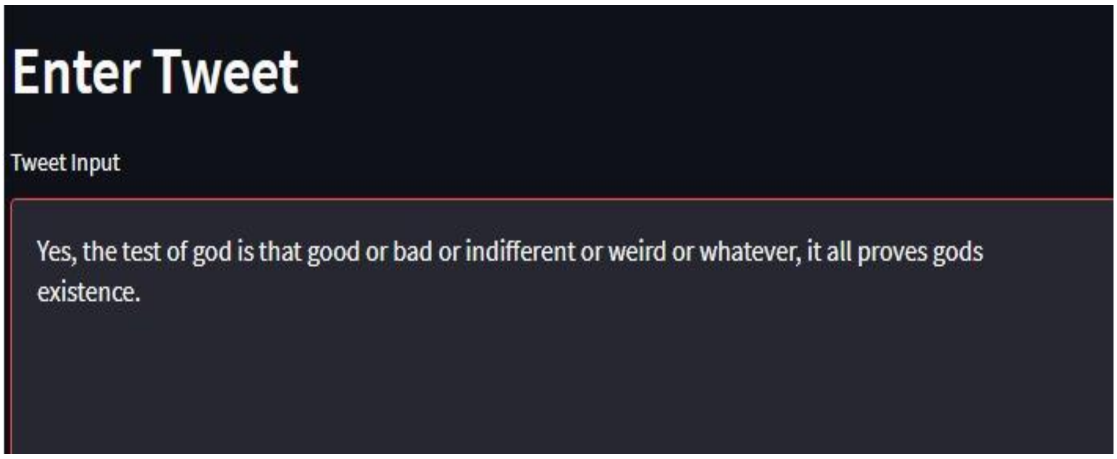


Fig 9.6 Ethnicity Cyberbullying output



Enter Tweet

Tweet Input

Yes, the test of god is that good or bad or indifferent or weird or whatever, it all proves gods existence.

Fig 9.7 Not Bullying Tweet Output

Fig 9.8 Not Bullying Output

CHAPTER-10

CONCLUSION

In conclusion, Cyber Saver harnesses the transformative power of machine learning, specifically utilizing the linear Support Vector Classification (SVC) algorithm, to effectively detect and combat the growing threat of cyberbullying. By analyzing both textual and image-based content, the model is capable of identifying subtle and complex patterns of harmful behaviour that might otherwise go unnoticed. The strength of linear SVC lies in its ability to efficiently classify high-dimensional data, making it particularly suitable for understanding the nuances of online communication. This enables Cyber Saver to not only respond to existing instances of cyberbullying but also to act as a preventive tool, flagging potentially harmful content before it escalates. Its implementation showcases how advanced machine learning techniques can be tailored to address real-world social challenges, contributing to a more secure and inclusive digital environment. As cyberbullying continues to evolve with technology, solutions like Cyber Saver represent a crucial step forward in using artificial intelligence for social good, promoting mental well-being and digital responsibility across online platforms.

CHAPTER-11

FUTURE ENCHANCEMENT

For future enhancements, "Cyber Saver" can be expanded by integrating more advanced machine learning techniques such as deep learning models, including Convolutional Neural Networks (CNNs) for image analysis and Recurrent Neural Networks (RNNs) or Transformers for nuanced text understanding. Incorporating Natural Language Processing (NLP) advancements like sentiment analysis, contextual embedding models and multilingual support can significantly improve detection accuracy, especially in recognizing sarcasm, slang, or coded language often used in cyberbullying. Additionally, real-time monitoring systems and adaptive learning mechanisms can be introduced to allow the model to evolve with new trends and threats as they emerge online. Another promising enhancement includes the development of a comprehensive reporting and alert system that not only notifies users or administrators but also provides resources for mental health support and guidance on responding to cyberbullying. Integrating user feedback loops and crowd-sourced moderation could further improve model performance and community trust. Moreover, ensuring ethical AI usage through bias mitigation, transparency in decision-making, and adherence to data privacy regulations will be crucial as the system scales.

REFERENCES

1. R. Kowalski, G. Giumetti, A. Schroeder and M. Lattanner, "Bullying in the digital age: A critical review and meta-analysis of cyberbullying research among youth", Psychological Bulletin, vol. 140, no. 4, pp. 1073-1137, 2014.
2. M. Gupta and S. Vaikole, "Recognition of Human Mental Stress Using Machine Learning Paradigms", SSRN Electronic Journal, 2020.
3. M. Raza, M. Memon, S. Bhatti and R. Bux, "Detecting Cyberbullying in Social Commentary Using Supervised Machine Learning", Advances in Intelligent Systems and Computing, pp. 621-630, 2020, [online] Available.
4. D. Soni and V. Singh, "See No Evil Hear No Evil", Proceedings of the ACM on Human-Computer Interaction, vol. 2, pp. 1-26, 2018.
5. T. Pradheep, J. Sheeba, T. Yogeshwaran and S. Pradeep Devaneyan, "Automatic Multi Model Cyber Bullying Detection from Social Networks", SSRN Electronic Journal, 2017.
6. R. Cohen et al., "Using computer technology to address the problem of cyberbullying", ACM SIGCAS Computers and Society, vol. 44, no. 2, pp. 52-61, 2014
7. B. Talpur and D. O’Sullivan, "Multi-Class Imbalance in Text Classification: A Feature Engineering Approach to Detect Cyberbullying in Twitter", Informatics, vol. 7, no. 4, pp. 52, 2020.
8. J. Hani, M. Nashaat, M. Ahmed, Z. Emad, E. Amer and A. Mohammed, "Social Media Cyberbullying Detection using Machine Learning", International Journal of Advanced Computer Science and Applications, vol. 10, no. 5, 2019.
9. A. Muneer and S. Fati, "A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter", Future Internet, vol. 12, no. 11, pp. 187, 2020.
10. Mifta Sintaha, Shahed Bin Satter, Niamat Zawad, Chaity Swamaker and Ahanaf Hassan, "Cyberbullying detection using sentiment analysis in social media", Core.ac.uk, 2021 [online] Available: https://core.ac.uk/reader/74351960.
11. A. Kumar, S. Nayak and N. Chandra, "Empirical Analysis of Supervised Machine Learning Techniques for Cyberbullying Detection", 2021.
12. M. Dadvar and F. de Jong, "Cyberbullying detection", Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion 2012, [online] Available.