A Project Report On

SENTIMENT ANALYSIS THROUGH NATURAL LANGUAGE PROCESSING

Submitted to partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

B. AJAY KUMAR (19911A0510)

Under the Esteemed Guidance of

Mr. Abdul Majeed

Assistant Professor

Department of Computer Science and Engineering

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE , Accredited by NAAC, NBA & permanently Affiliated to JNTUH)

Aziz Nagar Gate, C.B. Post, Hyderabad-500075

2022-2023

VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

(Approved by AICTE , Accredited by NAAC, NBA & permanently Affiliated to JNTUH)
Aziz Nagar Gate, C.B. Post, Hyderabad-500075

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report titled **"SENTIMENT ANALYSIS THROUGH NATURAL LANGUAGE PROCESSING"** is being submitted by **B. AJAY KUMAR (19911A0510)** in partial fulfillment for the award of the Degree of Bachelor of Technology in **Computer Science and Engineering,** is a record of bonafide work carried out by him under my guidance and supervision. These results embodied in this project report have not been submitted to any other University or Institute for the award of any degree of diploma.

Internal Guide
Mr. Abdul Majeed
Assistant Professor

Head of the Department
Dr. D Aruna Kumari
Professor

External Examiner

DECLARATION

I, **B. AJAY KUMAR.** Hereby declare that the project entitled, **“SENTIMENT ANALYSIS THROUGH NATURAL LANGUAGE PROCESSING”** submitted for the degree of Bachelor of Technology in Computer Science and Engineering is original and has been done by me and this work is not copied and submitted anywhere for the award of any degree.

Date: **B. AJAY KUMAR (19911A0510)**

Place: Hyderabad

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to the project guide, **Mr. Abdul Majeed**, Assistant Professor, department of CSE,Vidya Jyothi Institute of Technology, Hyderabad for his timely cooperation and valuable suggestions while carrying out this work. It is his kindness that made us learn more from him.

I'm grateful to **Dr. D Aruna Kumari**, Professor and HOD, department of CSE, for her help and support during the academic year.

I whole-heartedly convey my gratitude to **Dr. A. Padmaja,** Principal Vidya Jyothi Institute of Technology, Hyderabad for her constructive encouragement.

I would like to take this opportunity to express my gratitude to our Director
Dr. E. Sai Baba Reddy for providing necessary infrastructure to complete this project.

I would like to thank my parents and all the faculty members who have contributed to my progress through the course to come to this stage.

B. AJAY KUMAR (19911A0510)

ABSTRACT

SENTIMENT ANALYSIS USING NATURAL LANGUAGE PROCESSING

The project focuses on providing the type of sentiment behind every feedback and review that are written. We can analyze the nature of the feedback whether it is a positive-feedback or a negative-feedback by using Natural Language Processing. The following activities will take place:

- **Feedback as an Input:** We take the feedback of the user as an input.
- **Sentiment Analysis as an Output:** Through NLP, we can analyze the feedback and generate response.

Based on this analysis, if we want to make it widely accessible, we can create multiple phases through the analysis such as making the product improve as per customer's feedback and requirement which helps in gaining good demand for the product. We can also add several aspects like eyeball motion and also the physical signs that we make using hands through Natural Language Processing. Our sentiment analysis can solve the problem of physical approaches to know the feedback through physical phone calls or mouth talks and make sure to digitalize the manual services. For now, we are concentrating only on the type of sentiments and emotions behind the given feedback. The main components to build our sentiment analysis are:

- Python programming language
- Flask
- Natural language processing

The existing system has a rating-based system for the product's feedback and receiving mouth talks, a physical approach to customers to know the review of the product manually. The proposed system is the analysis using NLP and python to understand the level of demand and response that the customer provides through their feedback.

INDEX

LIST OF FIGURES

CHAPTER-1
INTRODUCTION

Opinions are central to almost all human activities because they are a key influence on people's behavior. Each time a decision needs to be made, humans look for others' opinions. In the real world, enterprises and organizations seek to know public opinion about their products and services. In turn, customers want to know others' opinions about a certain product before buying it. In the past, people looked for opinions from their friends and family, while organizations made polls or organized focus groups. Nevertheless, with the sudden growth of social networks such as Twitter and Facebook, individuals and organizations use data provided by these means to support their decision-making process. The field of sentiment analysis, also called opinion mining, emerged in this context.

Sentiment analysis is a relatively recent area in the field of data mining. There are different techniques for extracting, processing, and seeking objective data in texts. Nevertheless, there are subjective components that are also interesting. These components including opinions, sentiments, and emotions, among others, are the focus of sentiment analysis. Sentiment analysis includes a great number of tasks such as sentiment extraction and classification, subjectivity detection, opinion summary, and opinion spam detection, among others.

Sentiment analysis is an area with great development opportunities, particularly due to the huge growth of data available in the web, for example, in blogs, social networks, and forums, among others. One of the applications of opinion mining is product or service assessment by analyzing users' opinions or reviews. This application is highly important for organizations because it allows discovering what people think and say about a certain trademark. To review a product generally the organization is using the rating based system in which the customer will give the rating of the product with help of stars. If a customer gives one star then this means that the customer is dissatisfied with the product and five star meaning that they love the product. But with this we cannot know the drawback of the product or what the customer is loving about the product.

In this project we will be getting the customer reviews not based on rating system but he will give a description about his opinion of the product in four to five lines from which we will be analyzing the nature of the feedback whether it is a positive feedback or negative feedback using natural language processing.We will process the dataset and generate feedback of the dataset in the form of a confusion matrix. Confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an error matrix. It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

Millions of people are using social network sites to express their emotions, opinions and disclose about their daily lives. However, people write anything such as social activities or any comment on products. Through the online communities provide an interactive forum where consumers inform and influence others. Moreover, social media provides an opportunity for businesses for giving a platform to connect with their customers such as social media to advertise or speak directly to customers for connecting with the customer's perspective of products and services.

In contrast, consumers have all the power when it comes to what consumers want to see and how consumers respond.With this, the company’s success & failure is publicly shared and end up with word of mouth. However, social networks can change the behavior and decision making of consumers. For example, 87% of internet users are influenced in their purchase and decision by customer’s review. So that, if an organization can catch up faster on what their customers think, it would be more beneficial to organize to react on time and come up with a good strategy to compete with their competitors. Opinions are important to all human activities because they will be playing an important role to influence people’s opinions and behavior.Whenever we want to take certain decisions we will look for the opinions of other people, opinions of other people will be influencing our decisions.

In the real world, enterprises and organizations want to know public opinion and reviews about their products and services. In turn, customers want to know others' opinions while buying or doing anything hence the organization's product reviews will be based on the mouthtalk of the people. In the past, before buying or taking any new decisions they would seek advice or opinions from their family and friends. Organization made polls and groups to know their customer and public opinion’s. Nevertheless, with the sudden growth of social networks such as Twitter and Facebook, Instagram individuals and organizations, enterprises use these platforms to influence the people to buy their product and know the customer’s opinions and reviews so that they can improve their product more efficiently.

Sentiment analysis we will look at the emotions that are expressed in a text. Like Whether the text is positive or negative or neutral. It is also known as openmining or emotional artificial intelligence.Sentiment analysis is implemented using natural language processing and Applications of sentiment analysis are to analyze customer feedback, survey responses, and product reviews. For example, analyzing thousands of product reviews can generate useful feedback on your pricing or product features.

In sentiment analysis there are different techniques for performing extracting, processing, and seeking objective data in the reviews, texts. Sentiment analysis includes a number of tasks like loading and summarizing, segregating dataset, removing special characters, symbols from the text, extracting features, splitting data into test and train data sets, applying machine learning algorithms, predicting with test data and evaluating output from the confusion matrix.We can perform sentiment analysis with help of both supervised and unsupervised learning algorithms such as naives bayes, LSTM, SVM, Vader, TextBlob respectively.

Sentiment analysis is an area where there are huge opportunities for growth because data will be evolving day to day like we can see there would be a lot of data in social media and ecommerce platforms like facebook, twitter, instagram, amazon, flipkart,etc. One of the most widely used applications is analyzing the reviews of the customer from the social media platforms.

This will help a lot for the organizations and enterprises to know about people’s opinion,mindset, what do people think and what products they are expecting from the organizations and even organizations can know their drawbacks and correct their mistakes and reach a large number of people with the help of this.

Sentiment analysis is more trustworthy than human analysis. Human analysis will depend upon the customer's way of conveying to the organization person.While giving the review we will use tone, context and language and how the organization person understands that meaning depends on their experiences. For example, consider the following review regarding a product "Gets the job done, but it's not cheap!" There is both negative and positive sentiment in this sentence. Negative sentiment is about the price.

Positive sentiment is about the functionality of the product. But how can we now evaluate whether the sentence is positive or negative. Here we can say that sentiment analysis will evaluate the sentiment behind the text more efficiently and correctly than human based mouth talks. And this is an example of how subjectivity can influence sentiment perception. Sentiment analysis can be implemented using supervised and unsupervised learning algorithms which can analyze hundreds of megabytes of text in minutes. Instead of manually analyzing data in spreadsheets, we can do other valuable work.

CHAPTER-2

LITERATURE SURVEY

According to Brain Keith Norambeina who had researched sentiment analysis and opinion mining to scientific paper. It has been published that the scientific paper reviewing process is the main quality control mechanism for most communities which involves reviewing each paper in order to provide suggestions to authors for correcting and improving paper. Sentiment analysis and opinion mining is an area that has experienced considerable growth over the last decade. This area of research attempts to determine the feelings, opinions, emotions, among other things, of people on something or someone. To do this, natural language techniques and machine learning algorithms are used. This article discusses the problem of extracting sentiment and opinions from a collection of reviews on scientific articles conducted under an international conference on computing in northern Chile.

The first aim of this analysis is to automatically determine the orientation of a review and contrast this with the assessment made by the reviewer of the article. This would allow scientists to characterize and compare reviews crosswise and more objectively support the overall assessment of a scientific article. A hybrid approach that combines an unsupervised machine learning algorithm with techniques from natural language processing is proposed to analyze reviews. This method uses part-of-speech (POS) tagging to obtain the syntactic structure of a sentence. This syntactic structure, along with the use of dictionaries, allows determining the semantic orientation of the review through a scoring algorithm.

A set of experiments were conducted to evaluate the capability and performance of the proposed approaches relative to a baseline, using standard metrics, such as accuracy, precision, recall, and the F1-score. The results show improvements in the case of binary, ternary and a 5-point scale classification in relation to classical machine learning algorithms such as SVM and NB, but they also present a challenge to improve the multiclass classification in this domain. This paper aims to present the implementation of sentiment analysis methods in the area of scientific paper reviews as a proof of concept for future applications.

The used techniques include a Bayesian classifier (NB), a classifier built on the basis of support vector machines (SVM), an unsupervised classifier in the form of a scoring algorithm based on Part-Of-Speech and keyword matching, and finally a hybrid method using both the scoring algorithm and SVM. Methods used in opinion mining are related to data extraction and preprocessing, natural language processing, and machine learning methods, which play a fundamental role in the task of determining the orientation of an opinion. A learning task may be divided into two broad approaches: supervised learning, in which classes are provided in data, and unsupervised learning, in which classes are unknown and the learning algorithm needs to automatically generate class values. Supervised methods naïve Bayes and Support Vector Machines were used.

For the unsupervised learning task, an approach based on part-of-speech tagging and keyword matching was used. Furthermore, a hybrid approach which combines both supervised and unsupervised methods is proposed. Deep learning methods have not been tested due to the small size of the data set. While deep learning methods perform well in sentiment analysis, the number of parameters that must be estimated for deep learning to work well is too big for the amount of data present in this data set. Enlarging the data set is a difficult task since scientific reviews are an occluded genre and as such getting access to more data is not easy. Gathering more reviews has been left for future work, and given this, the application of deep learning methods on this data set has been left for future work.

CHAPTER-3
FEASIBILITY STUDY

Feasibility Study in Software Engineering is a study to evaluate feasibility of proposed project or system. Feasibility study is one of the important four stages of the Software Project Management Process. As the name suggests, feasibility study is the feasibility analysis or it is a measure of the software product in terms of how beneficial product development will be for the organization from a practical point of view. Feasibility study is carried out based on many purposes to analyze whether a software product will be right in terms of development, implantation, contribution of project to the organization etc. To evaluate the project we cover the feasibility of the project from a technological, economical and legal perspective. Those perspectives would help us have a broad vision on the requirements and implications related to the project.

In order to evaluate if the project can be done in the given time frame, we are using the TEL-evaluation methods, where we cover the feasibility of the project from a technological, economical and legal perspective. Those perspectives would help us have a broad vision on the requirements and implications related to the project. We also discuss in this section the methodology used in conducting the project.

Need of Feasibility Study : Feasibility study is so important stage of Software Project Management Process as after completion of feasibility study it gives a conclusion of whether to go ahead with proposed project as it is practically feasible or to stop proposed project here as it is not right/feasible to develop or to think/analyze about proposed project again. Along with this Feasibility study helps in identifying risk factors involved in developing and deploying systems and planning for risk analysis also narrows the business alternatives and enhances success rate analyzing different parameters associated with proposed project development.

Below are some key benefits of conducting a feasibility study:

- Improves project teams' focus
- Identifies new opportunities
- Provides valuable information for a "go/no-go" decision
- Narrows the business alternatives
- Identifies a valid reason to undertake the project
- Enhances the success rate by evaluating multiple parameters
- Aids decision-making on the project

3.1 Technological Feasibility

This project had been developed using technologies and libraries pertinent to the Natural language toolkit, Python programming language, Machine learning- classification algorithms. This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems.

Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building currently, this project is not technically feasible.

3.2 Economical Feasibility

This project will be based on Free and Open Source Technologies and Libraries that are readily available to developers and scientists, free of cost. This means that we don’t have to worry about costs related to licensing or reusing source code and that the only costs related to the project are related to the time and the effort spent into developing it.This means that we don’t have to worry about costs related to licensing or reusing source code and that the only costs related to the project are related to the time and the effort spent into developing it.

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3.3 Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let’s say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization’s ideal location isn’t zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

3.4 Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether and how well the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

CHAPTER-4

SYSTEM REQUIREMENTS

4.1 Existing System

As we mentioned previously, today there exists a rating based system, with the help of it organizations would be knowing their product feedback or with mouth talk we can know the product reviews and opinions of other people. Millions of people are using social network sites to express their emotions, opinions and disclose about their daily lives. Moreover, social media provides an opportunity for businesses for giving a platform to connect with their customers such as social media to advertise or speak directly to customers for connecting with the customer's perspective of products and services.Social media will influence many people in order to know review currently rating system is used but with help of it we can’t know the sentiment behind the reviews.

4.2 Proposed System

We have successfully proposed the “Sentiment analysis using natural language processing” for replacing the rating based system and mouth talks used for reviewing the product. This application is flexible and can easily be accessed by anyone. So that user can know the sentiment behind the reviews or any data set without using a rating based system or mouth talks. We will be giving the reviews of the users as an input file.

It will process the input data and clean data and analyze the sentiments behind the reviews and give the sentiment in the form of a confusion matrix.We will collect the reviews of the users from multiple social media sites such as facebook, twitter, instagram, etc.The input file will be in the form of .csv file and after processing the input we will be getting output in the form of a confusion matrix and we will also know the best algorithm among the two we have used and get the accuracy of the model which is known as the score of the model.

4.3 System Requirements

4.3.1 Software Requirements

- OS : Windows, Linux (With any web browser)
- Programming languages and libraries : Natural Language toolkit, Sklearn, Numpy,Python

4.3.2 Hardware Requirements

- RAM : 4GB and Higher
- Processor : Intel i5 or above
- Hard Disk : 16 GB or above

4.4 Requirements Definition

After the severe continuous analysis of the problems that arose in the existing system, we are now familiar with the requirements that are required by the current system. The requirements that the system needs are categorized into the functional and nonfunctional requirements. These requirements are listed below:

4.4.1 Functional Requirements

Functional requirements define which functions or features that are to be incorporated in any system to fulfill the business requirements and to be acknowledged by the clients. On the premise, the functional requirements specify the relationship between the inputs and outputs. All the operations to be performed on the input data to obtain output are to be specified. This includes specifying the validity checks on the input and output data, parameters affected by the operations and the other operations, which must be used to transform the inputs into outputs. Functional requirements specify the behavior of the system for valid input and outputs.

Functional requirements deal with the functionality of the software in the engineering view. The component flow and the structural flow of the same is enhanced and described by it.

The functional statement deals with the raw datasets that are categorized and learning from the same dataset. Later the datasets are categorized into clusters and the impairment of the same is checked for the efficiency purpose. After the dataset cleaning the data is cleansed and the machine learns and finds the pattern set for the same it undergoes various iterations and produces output.

Functional requirement defines which functions or features that are to be incorporated in any system to fulfill the business requirements and to be acknowledged by the clients. On the premise, the functional requirements specify the relationship between the inputs and outputs. All the operations to be performed on the input data to obtain output are to be specified. This includes specifying the validity checks on the input and output data, parameters affected by the operations and the other operations, which must be used to transform the inputs into outputs. Functional requirements specify the behavior of the system for valid input and outputs.

Functional Requirement is a description of the service that the software system or its component. A Function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

These are the requirements that the end user specifically demands as basic facilities that the end user specifically demands as the basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

4.4.2 Non Functional Requirements

Non-functional requirements provide a description of features, characteristics and capacity of the system and furthermore it may constrain the boundaries of the proposed system.

The following are the non-functional requirements that are essential depending on the performance, cost, control and gives security efficiency and services.

Based on the above explained non-functional prerequisites are as follows:

- User-friendly
- System should provide better accuracy
- To perform efficiently with better throughput and response time
- A non-functional requirement tells us about the system's behavior.
- This also specifies how are the system's quality characteristics or quality attributes.
- The system is highly reliable.
- Resource consumption is quite low.
- We can add more resources to our project without disturbing the current scenario.

Non-Functional Requirements address vital issues of quality for software systems.

- **Security**: Secure connection should be established for transmission of any data. The system's back- end servers shall only be accessible to authenticated management.
- **Performance**: Any page that the user tries to access should load in less than 2 seconds.
- **Maintainability**: In case of a failure, a re-initialization of the system will be done. Also, the software design is being done with modularity in mind so that maintainability can be done efficiently.
- **Usability**: Users can easily navigate its interface and can easily determine what a feature is and what it can do. Non-technical background of a user should not be an obstacle to understanding and using the system.
- **Availability**: The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs.
- **Browser Compatible**: The application should be accessible through all latest web browsers such as Google Chrome and Internet Explorer.

CHAPTER-5
SYSTEM DESIGN

5.1 UML Diagrams

UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modeling tools include. In its simplest form, a use case can be described as a specific way of using the system from a User's (actor's) perspective. A more detailed description might characterize a use case as:

- a pattern of behavior the system exhibits
- a sequence of related transactions performed by an actor and the system
- delivering something of value to the actor

Use cases provide a means to:

- capture system requirements
- communicate with the end users and domain experts
- Test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system. Since all the needs of a system typically cannot be covered in one use case, it is usual to have a collection of use cases. Together this use case collection specifies all the ways of using the system.

A UML system is represented using five different views that describe the system from a distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

User Model View

- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-user's perspective.

Structural model view

- In this model the data and functionality are arrived from inside the system.
- This model view models the static structures.

Behavioral Model View

- It represents the dynamic of behavior as parts of the system, depicting the interactions of collection between various structural elements described in the usermodel and structural model view.

Implementation Model View

- In this the structural and behavioral parts of the system are represented as they are to be built.

Environmental Model View

- In this the structural and behavioral aspect of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioral

5.1.1 Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

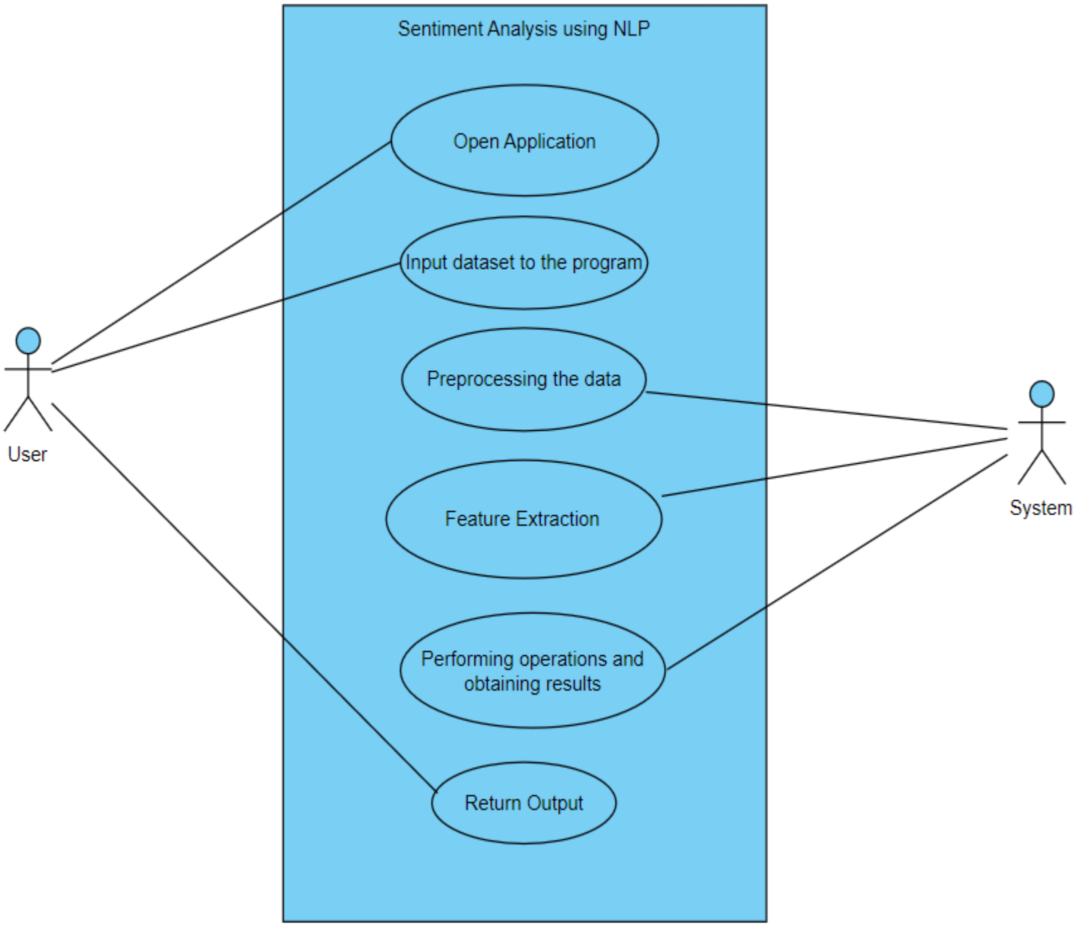

Sentiment Analysis using NLP
Open Application
Input dataset to the program
Preprocessing the data
Feature Extraction
Performing operations and obtaining results
Return Output
User
System

Fig. 5.1.1 Use case diagram

Our use case diagram describes the use case between the sentiment analysis system and the user which displays the appropriate functions whenever performed by both the user and system simultaneously. Everything works in the given flow of the diagram.

5.1.2 Class Diagram

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

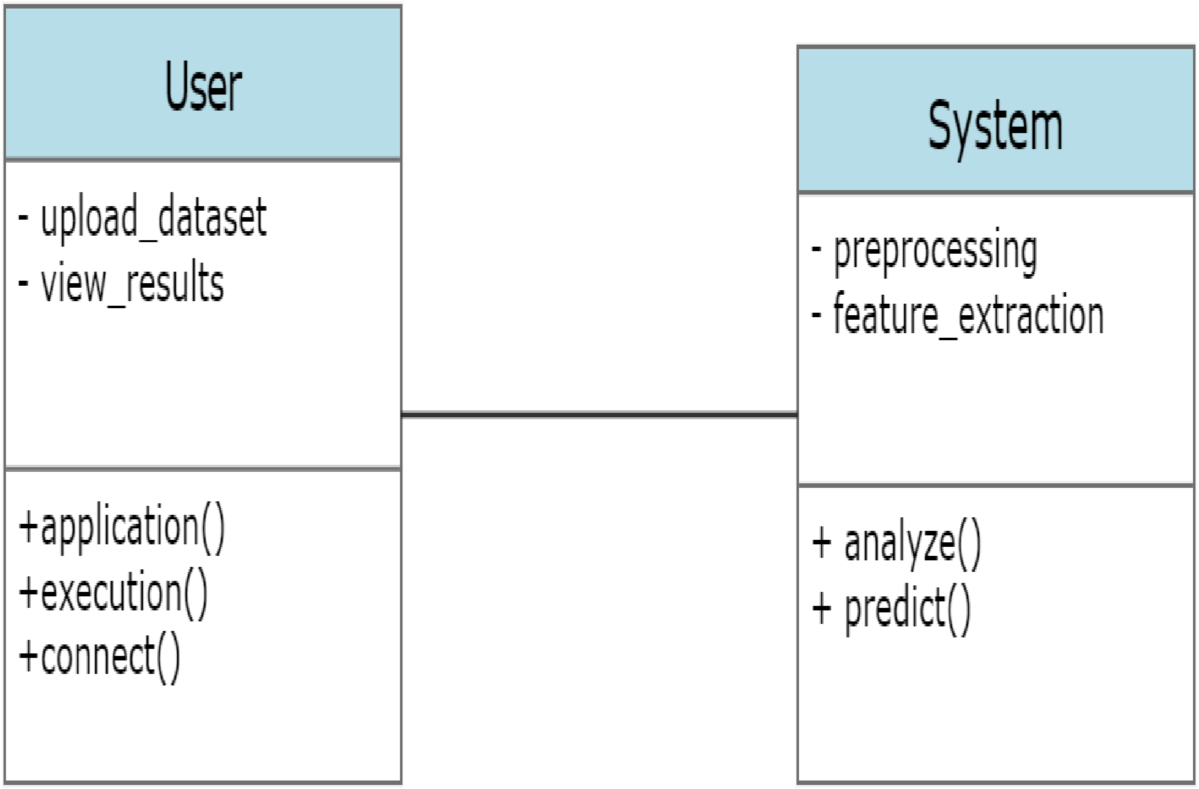


Fig 5.1.2 Class diagram

Our class diagram in the project describes two types of classes used in the project. Each class has its own significance and performs the given operations as per the defined attributes. We have defined two classes here which are relevant to the project implementation.

5.1.3 Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

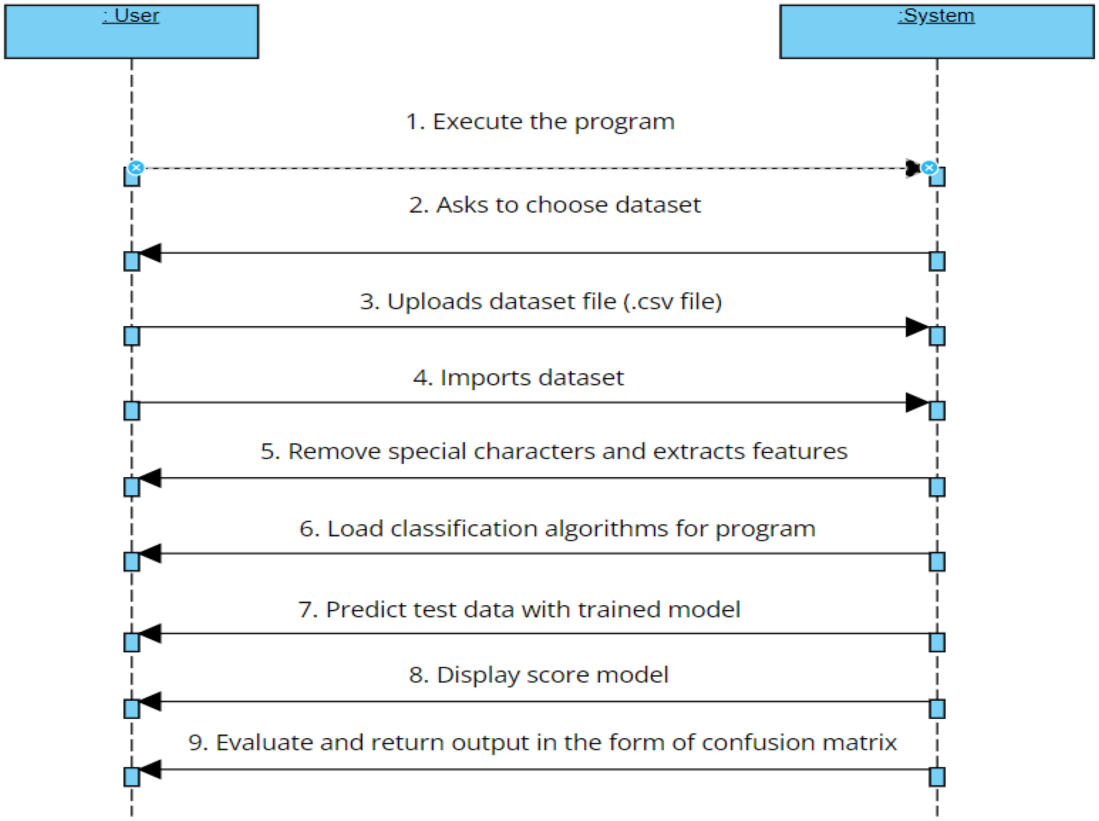

: User
:System
1. Execute the program
2. Asks to choose dataset
3. Uploads dataset file (.csv file)
4. Imports dataset
5. Remove special characters and extracts features
6. Load classification algorithms for program
7. Predict test data with trained model
8. Display score model
9. Evaluate and return output in the form of confusion matrix

Fig. 5.1.3 Sequence Diagram

The sequence diagram in our project defines the flow of operations in an ordered manner. The sequence of functions is followed between both the user and the system. It displays how the game is connected with the user and the breath detection is taking place by taking users chest movements.

5.1.4 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

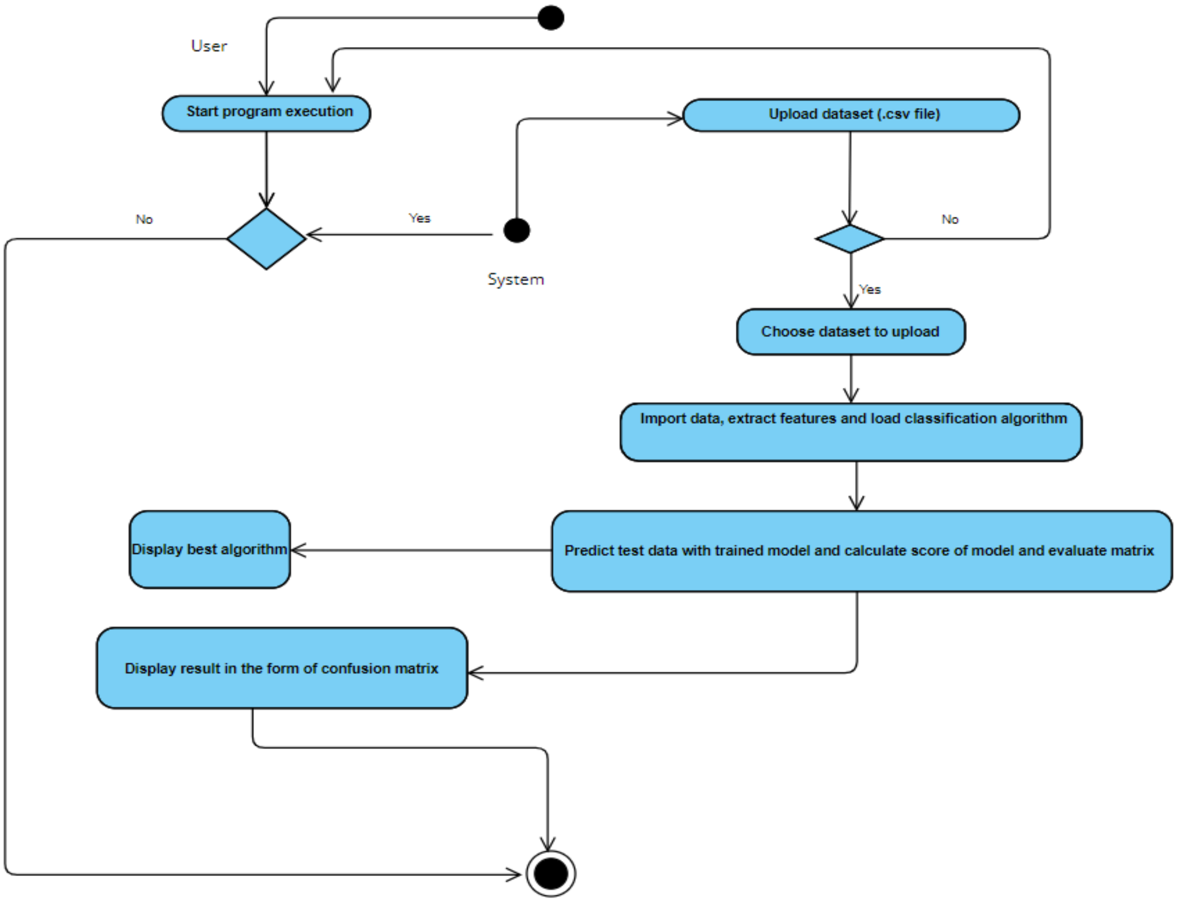

User
Start program execution
Upload dataset (.csv file)
No
Yes
System
No
Yes
Choose dataset to upload
Import data, extract features and load classification algorithm
Display best algorithm
Predict test data with trained model and calculate score of model and evaluate matrix
Display result in the form of confusion matrix

Fig. 5.1.4 Activity Diagram

The activity diagram depicts the types of activities that are ongoing in the project. They are utmost helpful in providing the necessary gateways for the user and the system. It also describes the flow between each activity having its own utility.

5.1.5 Data Flow Diagram

The flow of data of a system is represented as Data Flow Diagram. It also gives insight of inputs and outputs of each entity and the process itself.It does not have control flow and no loops or decision rules are present.

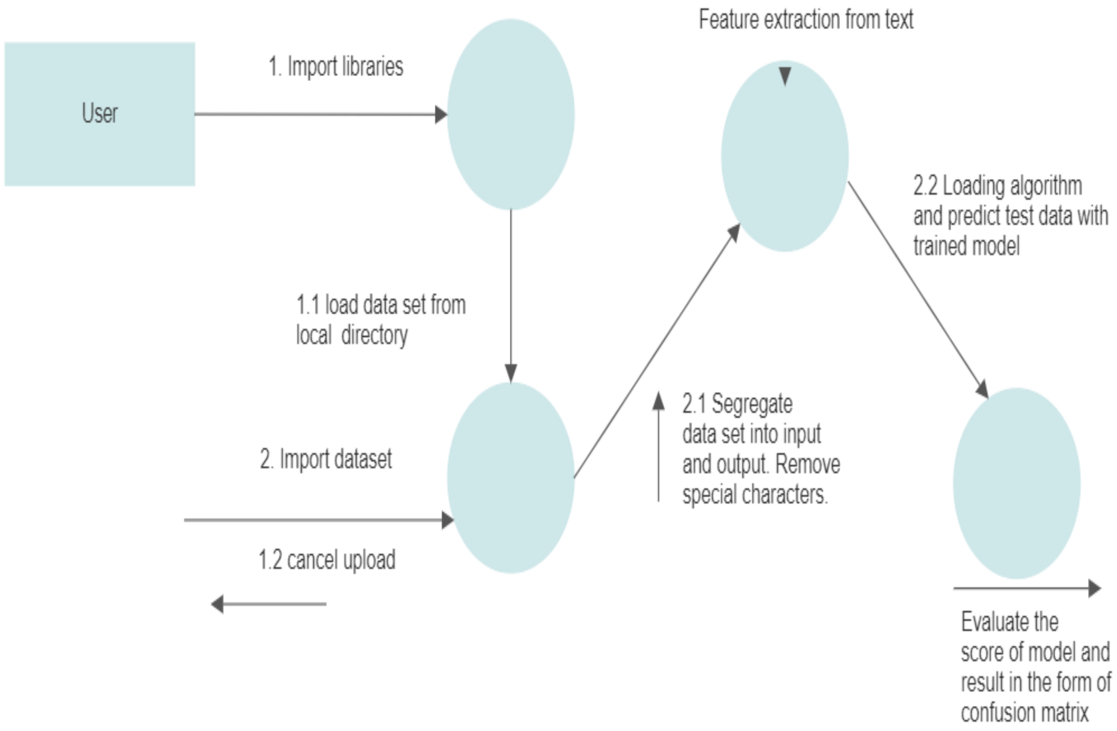

User
1. Import libraries
Feature extraction from text
2.2 Loading algorithm and predict test data with trained model
1.1 load data set from local directory
2.1 Segregate data set into input and output. Remove special characters.
2. Import dataset
1.2 cancel upload
Evaluate the score of model and result in the form of confusion matrix

Fig. 5.1.5 Data flow diagram

The data flow diagram in our project shows how the data input and data output coordinates to give proper functionality of the project. They are widely used to provide efficient performance of the project while having a good running environment.

5.2 Software Development Life Cycle (SDLC)

The software development life cycle (SDLC) is a process by which software is developed and deployed. It's a process that encompasses every phase of software creation, from conception to maintenance after the software is released.

5.2.1 Iterative Model

In the Iterative model, the iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

In this Model, we can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.

The Iterative Model allows accessing earlier phases, in which the variations are made respectively. The final output of the project was renewed at the end of the Software Development Life Cycle (SDLC) process.

Advantages of Iterative Model:

1. Testing and debugging during smaller iterations is easy.
2. A Parallel development can be planned.
3. It is easily acceptable to the ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

CHAPTER-6
SOFTWARE ENVIRONMENT

6.1 Technologies used in the application

6.1.1 Google Colab

Google Colab was developed by Google to provide free access to GPU’s and TPU’s to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook.
Programming Languages are an intermediate form between human-understandable language and machine understandable language. Every application is built using one of the many programming languages available. Maybe a person with a computer science background can understand, but not everyone can.

Remember, as Software Developers, we develop applications for people with little computer science knowledge.Consider you are creating a machine learning model to improve customer satisfaction for a local store, in that case you will have to explain how the model can do this task, and you can't just explain him with your code base. Most people facing this situation will prepare a separate presentation. Notebooks were created so that it is not necessary.

Notebook documents can include executable lines of code along with text, images, figures, tables, graphs, equations, and much more graphical data. In simple words, Notebook documents are a way of creating human-readable executable documents.Colab is a Jupyter Notebook-like product from Google Research. A Python program developer can use this notebook to write and execute random Python program codes just using a web browser. In a nutshell, Colab is a cloud-hosted version of Jupyter Notebook. To use Colab, you do not need to install and run time or upgrade your computer hardware to meet Python’s CPU/GPU intensive workload requirements. Furthermore, Colab gives you free access to computing infrastructure like storage, memory, processing capacity, graphics processing units (GPUs), and tensor processing units (TPUs).

Special Features of Google Colab:

- GPUs and TPUs

  Free Colab users get chargeless access to GPU and TPU runtimes for up to hours. Its GPU runtime comes with Intel Xeon CPU @2.20 GHz, 13 GB RAM, Tesla K80 accelerator, and 12 GB GDDR5 VRAM.The TPU runtime consists of an Intel Xeon CPU @2.30 GHz, 13 GB RAM, and a cloud TPU with 180 teraflops of computational power.With Colab Pro or Pro+, you can commission more CPUs, TPUs, and GPUs for more than 12 hours.

- Notebook Sharing

  Python code notebook has never been accessible before Colab. Now, you can create shareable links for Colab files that are saved on your Google Drive. Now, share the link with the collaborator who wants to work with you. Moreover, you can also invite programmers to work with you using Google emails.

- Special Library Installation

  Colab lets you install non-Colaboratory libraries (AWS S3, GCP, SQL, MySQL, etc.) that are unavailable in the Code snippets. All you need to do is add a one-liner code with following code prefixes.

  !pip install (example: !pip install matplotlib-venn)

- Pre Installed Libraries

  Google Colab offers multiple pre-installed libraries so that you can import the required library from Code snippets.Such libraries include NumPy, Pandas, Keras, Matplotlib, PyTorch, TensorFlow, Keras, and more ML libraries.

- Collaborative Coding

  Co-coding is indispensable for group projects. It helps your team to complete milestones earlier than the expected time frame. If your team needs real-time collaboration on ML and data science projects, Google Collaborative is just the tool.Simply send an editable link with the collaborators or invite collaborators for group coding. The entire Python notebook updates automatically as the team codes, and you get the feeling of working on shared Google Sheets or Docs.

- Multiple Data Sources

  Google Colaboratory supports various data sources for your ML and AI-training projects For example, you can import data from a local machine,mount Google Drive to a Colab instance, fetch remote data,and clone GitHub repo into Colab.

Reasons for using Google Colab:

- Interactive tutorials to learn machine learning and neural networks.
- Write and execute Python 3 code without having a local setup.
- Execute terminal commands from the Notebook.
- Import datasets from external sources such as Kaggle.
- Save your Notebooks to Google Drive.
- Import Notebooks from Google Drive.
- Free cloud service, GPUs and TPUs.
- Integrate with PyTorch, Tensor Flow, Open CV.
- Import or publish directly from/to GitHub.

6.1.2 Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for web development (server-side),software development, mathematics,system scripting.Python can be used on a server to create web applications. It can be used alongside software to create workflows.It can connect to database systems. It works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker.

In January 2019, active Python core developers elected a five-member Steering Council to lead the project. Python 2.0 was released on 16 October 2000, with many major new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released on 3 December 2008, with many of its major features backported to Python 2.6.x and 2.7.x. Releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3. Python 2.7's end-of-life was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.

No further security patches or other improvements will be released for it. Currently only 3.7 and later are supported. In 2021, Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues leading to possible remote code execution and web cache poisoning. In 2022, Python 3.10.4 and 3.9.12 were expedited and 3.8.13, and 3.7.13, because of many security issues. When Python 3.9.13 was released in May 2022, it was announced that the 3.9 series (joining the older series 3.8 and 3.7) would only receive security fixes in the future.

Python is a very popular general-purpose interpreted, interactive, object-oriented, and High-level programming language.Python is a dynamically-typed and garbage collected Programming language.It was created by Guido van Rossum during 1985-1990.Like Perl, Python source code is also available under the GNU General Public License (GPL). It supports functional and structured programming methods as well as OOP. It can be used as a scripting language or can be compiled to byte-code for building large applications.It provides very high-level dynamic data types and supports dynamic type checking.It supports automatic garbage collection.It can be easily integrated with C,C++,COM.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming is fully supported, and many of their features support functional programming and aspect- oriented programming (including metaprogramming and meta objects) Many other paradigms are supported via extensions,including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and cycle-detecting garbage collector for memory management. It uses dynamic name resolution (late binding),which binds method and variable names during program execution. Its design offers some support for functional programming in the Lisp tradition. It has filter map and reduce functions list comprehensions, dictionaries,sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Its core philosophy is summarized in the document The Zen Of Python(PEP 20), which includes aphorisms such as :

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.

Rather than building all of its functionality into its core, Python was designed to be highly extensible via modules. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from this frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology.In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one and preferably only one obvious way to do it" philosophy. Alex Martelli, a Fellow at Python Software.

Characteristics of Python

Following are important characteristics of Python Programming −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

6.1.3 NumPy

NumPy is a Python library used for working with arrays.NumPy (Numerical Python) is an open source Python library that’s used in almost every field of science and engineering. It’s the universal standard for working with numerical data in Python, and it’s at the core of the scientific Python and PyData ecosystems. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development.

The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.The NumPy library contains multidimensional array and matrix data structures (you’ll find more information about this in later sections). It provides ndarray, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

NumPy gives you an enormous range of fast and efficient ways of creating arrays and manipulating numerical data inside them. While a Python list can contain different data types within a single list, all of the elements in a NumPy array should be homogeneous. The mathematical operations that are meant to be performed on arrays would be extremely inefficient if the arrays weren’t homogeneous.

NumPy arrays are faster and more compact than Python lists. An array consumes less memory and is convenient to use. NumPy uses much less memory to store data and it provides a mechanism of specifying the data types. This allows the code to be optimized even further. The NumPy library contains multidimensional array and matrix data structures (you’ll find more information about this in later sections). It provides ndarray, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays.

It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.NumPy gives you an enormous range of fast and efficient ways of creating arrays and manipulating numerical data inside them. While a Python list can contain different data types within a single list, all of the elements in a NumPy array should be homogeneous. The mathematical operations that are meant to be performed on arrays would be extremely inefficient if the arrays weren't homogeneous. NumPy arrays are faster and more compact than Python lists. An array consumes less memory and is convenient to use. NumPy uses much less memory to store data and it provides a mechanism of specifying the data types. This allows the code to be optimized even further.

Essentially, C and Fortran orders have to do with how indices correspond to the order the array is stored in memory. In Fortran, when moving through the elements of a two-dimensional array as it is stored in memory, the first index is the most rapidly varying index. As the first index moves to the next row as it changes, the matrix is stored one column at a time. This is why Fortran is thought of as a Column-major language. In C on the other hand, the last index changes the most rapidly. The matrix is stored by rows, making it a Row-major language. What you do for C or Fortran depends on whether it's more important to preserve the indexing convention or not reorder the data.

NumPy does not require any external linear algebra libraries to be installed. However, if these are available, NumPy's setup script can detect them and use them for building. A number of different LAPACK library setups can be used, including optimized LAPACK libraries such as OpenBLAS or MKL. The choice and location of these libraries as well as include paths and other such build options can be specified in a site.cfg file located in the NumPy root repository or a .numpy-site.cfg file in your home directory. See the site.cfg.example example file included in the NumPy repository or for documentation, and below for specifying search priority from environmental variables.

6.1.4 Sklearn

Scikit-learn is an open source data analysis library, and the gold standard for Machine Learning (ML) in the Python ecosystem. Key concepts and features include:

- Algorithmic decision-making methods, including:
  - **Classification:** identifying and categorizing data based on patterns
  - **Regression:** predicting or projecting data values based on the average mean of existing and planned data.
  - **Clustering:** automatic grouping of similar data into datasets.
- Algorithms that support predictive analysis ranging from simple linear regression to neural network pattern recognition.
- Interoperability with NumPy, pandas, and matplotlib libraries.

ML is a technology that enables computers to learn from input data and to build/train a predictive model without explicit programming. ML is a subset of Artificial Intelligence (AI). Whether you are just looking for an introduction to ML, want to get up and running fast, or are looking for the latest ML research tool, you will find that scikit-learn is both well-documented and easy to learn/use. As a high-level library, it lets you define a predictive data model in just a few lines of code, and then use that model to fit your data. It's versatile and integrates well with other Python libraries, such as matplotlib for plotting, numpy for array vectorization, and pandas for dataframes.
To use scikit-learn, you should first be familiar with some of the terminology typically used in ML projects.

French research scientist David Cournapeau's scikits.learn is a Google Summer of Code venture where the scikit-learn project first began. Its name refers to the idea that it's a modification to SciPy called "SciKit" (SciPy Toolkit), which was independently created and published. Later, other programmers rewrote the core codebase.

The French Institute for Research in Computer Science and Automation at Rocquencourt, France, led the work in 2010 under the direction of Alexandre Gramfort, Gael Varoquaux, Vincent Michel, and Fabian Pedregosa. On February 1st of that year, the institution issued the project's first official release. In November 2012, scikit-learn and scikit-image were cited as examples of scikits that were "well-maintained and popular". One of the most widely used machine learning packages on GitHub is Python's scikit-learn.

Scikit-learn is mainly coded in Python and heavily utilizes the NumPy library for highly efficient array and linear algebra computations. Some fundamental algorithms are also built in Cython to enhance the efficiency of this library. Support vector machines, logistic regression, and linear SVMs are performed using wrappers coded in Cython for LIBSVM and LIBLINEAR, respectively. Expanding these routines with Python might not be viable in such circumstances.

Scikit-learn works nicely with numerous other Python packages, including SciPy, Pandas data frames, NumPy for array vectorization, Matplotlib, seaborn and plotly for plotting graphs, and many more.

- Accuracy – the fraction of predictions that a classification model got right.

  In multi-class classification, accuracy is defined as follows:

  Accuracy = Correct Predictions / Total Number Of Examples

- In binary classification, accuracy has the following definition:

  Accuracy * = (True Positives + True Negatives) / Total Number Of Examples

- Example Data – particular instance (feature) of data, defined as x. There are two categories of data examples:

  Labeled Data– includes both feature(s) and the label, defined as:

  {features, label}: (x, y)

Unlabeled Data – contains features but not the label, defined as:

{features, ?}: (x, ?)

- Feature – an input variable. It is a measurable characteristic or property of a thing being observed. Every ML project has 1 or more features.
- Clustering – a technique that groups data points based on their similarities. Each group is called a Cluster.
- K-Means Clustering – an unsupervised learning technique that looks for a fixed number (k) of means (centroids) of data points, and assigns them to the nearest cluster.
- Model – defines the relationship between features and a label. For example, a Rumor Detection Model that associates certain features associated with rumors.
- Regression vs Classification – both are models that allow you to make predictions that answer questions, such as which team will win a sporting event.
  Regression models provide a numerical or continuous value.
  Classification models provide a categorical or discrete value.
- Supervised Learning – the algorithm uses a labeled dataset to 'learn' how to recognize correct answers, which it can then apply to training data. The algorithm's accuracy is then evaluated and refined. Most ML projects use supervised learning.
- Unsupervised Learning – the algorithm tries to make sense of unlabeled data by 'learning' features and patterns on its own.

For computers to learn without being explicitly programmed, algorithms are required. Algorithms are merely sets of rules applied to computation. The extensive community of open-source programs is one of the key justifications for using them, and Sklearn is comparable in this regard.

There have been roughly 35 contributors to Python's scikit-learn library, with Andreas Mueller being the most noteworthy. On the scikit learn the main page, many Organizations, including Evernote, Inria, and AWeber, are listed as customers. But the actual utilization is much higher than that. ML algorithm basic concepts:

- **Representation** – is a way to configure data such that it can be assessed. Examples include decision trees, sets of rules, instances, graphical models, neural networks, support vector machines, model ensembles and others.

- **Evaluation** – given a hypothesis, evaluation is a way of assessing its validity. Examples include accuracy, prediction and recall, squared error, likelihood, posterior probability, cost, margin, entropy k-L divergence and others.

- **Optimization –** the process of adjusting hyperparameters in order to minimize model errors by using techniques like combinatorial optimization, convex optimization, constrained optimization, etc.

Scikit - Learn ML Algorithms :

- Linear Regression Algorithm **:** Linear Regression is a supervised ML algorithm in which the predicted output is a slope in a straight line. It’s used to predict values within a given set of data points and not beyond. Simple linear regression uses the slope-intercept form of a straight line y=mx+b where m and b are variables that the algorithm uses to learn or create the most accurate predictive slope, x represents input data, y represents the prediction.

- Decision Tree Algorithm : A Decision Tree algorithm formulates a tree composed of root nodes (points where a choice must be made), branch nodes (binary yes/no answers to the choice) and leaf nodes (represent variables).

- Random Forest : A Random Forest is a model composed of multiple Decision Trees and different learning algorithms (ensemble learning method) to obtain better predictive analysis than could be obtained from any single learning algorithm.

6.1.5 Pandas

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's ActivePython. Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data, including:

- Data cleansing
- Data fill
- Data normalization
- Merges and joins
- Data visualization
- Statistical analysis
- Data inspection
- Loading and saving data

In fact, with Pandas, you can do everything that makes world-leading data scientists vote Pandas as the best data analysis and manipulation tool available. Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way towards this goal.

Here are just a few of the things that pandas does well:

- Easy handling of missing data (represented as NaN, NA, or NaT) in floating point as well as non-floating point data

- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects

- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations

- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data

- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects

- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets

- Intuitive merging and joining data sets

- Flexible reshaping and pivoting of data sets

- Hierarchical labeling of axes (possible to have multiple labels per tick)

- Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving/loading data from the ultrafast HDF5 format

- Time series-specific functionality: date range generation and frequency conversion, moving window statistics, date shifting and lagging

6.1.6 NLTK

Natural Language Processing (NLP) is a process of manipulating or understanding the text or speech by any software or machine. An analogy is that humans interact and understand each other’s views and respond with the appropriate answer. In NLP, this interaction, understanding, and response are made by a computer instead of a human.

NLTK (Natural Language Toolkit) Library is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response.

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK 3.

NLP Library	**Description**
NLTK	This is one of the most usable and mother of all NLP libraries.
spaCy	This is a completely optimized and highly accurate library widely used in deep learning
Stanford CoreNLP Python	For client-server-based architecture, this is a good library in NLTK. This is written in JAVA, but it provides modularity to use it in Python.
TextBlob	This is an NLP library which works in Python2 and python3. This is used for processing textual data and provides mainly all types of operation in the form of API.
Gensim	Gensim is a robust open source NLP library supported in Python. This library is highly efficient and scalable.
Pattern	It is a light-weighted NLP module. This is generally used in Web-mining, crawling or such a spidering task to provide mainly all types of operation in the form of API.
Polyglot	For massive multilingual applications, Polyglot is the best suitable NLP library. Feature extraction in the way of Identity and Entity.
PyNLPI	PyNLPI is also known as ‘Pineapple’ and supports Python. It provides a parser for many data formats like FoLiA/Giza/Moses/ARPA/Timbl/CQL
Vocabulary	This library is best to get Semantic type information from the given text

CHAPTER-7
IMPLEMENTATION

7.1 Overview

Sentiment analysis through natural language processing will detect the positive, negative and neutral statements from the dataset provided as input.Application is designed in such a way that whenever user uploads the data as input file then sentiment behind the statements is evaluated in the form of confusion matrix. With the help of a confusion matrix it is easy to determine the performance of classification models for the given set of data. It can be predicted only if the true values are known.

In this process the first step is to provide the dataset as input and load,summarize the dataset then segregation dataset after which we will remove special characters,symbols with regular expressions and stopwords then extract features we will now split dataset into two datasets one is test data and another train data set now load machine learning algorithm and apply on the dataset and predict with test data and evaluate the result from confusion matrix. The proposed methodology in this project involves using a machine learning - based natural language processing approach with advanced machine learning algorithm techniques for sentiment analysis.

We also make use of two classification algorithms which are used in the sentiment analysis of the uploaded dataset, we use random forest classifier algorithm and naive bayes algorithm. We use both of these algorithms to predict the sentimental emotions behind the feedback dataset. Both of them have their own pros and cons but for our project, the random forest classifier algorithm is best used. By comparing the efficiency score of the algorithm we can understand that the random forest classifier has the best algorithm.

The methodology consists of the following steps:

- **Dataset preparation:** A large collection of labeled feedback from e-commerce applications like amazon, twitter, flipkart is used to train and validate the deep learning model. The dataset is preprocessed and augmented using various techniques such as vectorization, nltk libraries and machine learning algorithms like random classifier, bayes` naive algorithms to increase the diversity and quantity of data.

- **Model architecture selection:** A suitable machine learning model architecture, specifically natural language processing, is selected for the analysis. The NLP using machine learning can learn discriminative features directly from the input data, which can improve the accuracy and robustness of the model.

- **Model training:** The selected function flow of analysis is trained on the prepared dataset. The training process involves optimizing the model parameters to analyze the sentiment using random classifiers and vectorization algorithms. The training is performed in multiple epochs until the validation loss converges.

- **Model evaluation:** The trained model is evaluated on a separate test set to measure its performance in terms of accuracy, precision, recall. The performance metrics are compared with existing state-of-the-art methods.

- **Sentiment Analysis :** The trained model is used to analyze and classify the appropriate sentiment for the given dataset. The input dataset is preprocessed using the same techniques used during the dataset preparation phase, and the model predicts the appropriate sentiment through the confusion matrix.

7.2 Implementation Steps

- The first step in the implementation is to import the libraries from nltk which we will be using in evaluating the sentiment behind the sentiment.

- The second step in the implementation would be uploading the file from the local directory.

- The third step in the implementation would be importing the data set which we have uploaded as the input file from the local directory.We can check whether the data is imported correctly or not by printing the data.

- The fourth step in the implementation would be segregating the input data into input and outputs. Basically the input data is the 3 months data of the twitter reviews which will be collecting the user's twitter id, user’s names, user’s location, and time zone, negative reason, negative confidence, tweet created.From this input data we will segregate into input data and output data.

- The fifth step in the implementation would be removing special characters from the text.We remove HTML tags, URLs and non-alphanumeric characters from the dataset using regex functions. Stopwords (commonly used words like ‘and’, ‘the’, ‘at’ that do not hold any special meaning in a sentence) are also removed from the corpus using the nltk stopwords.

- The sixth step in the implementation would be extracting the features from the text.

- The seventh step in the implementation would be the loading algorithm. Here in our project we are using two machine learning algorithms i.e naive bayes and random forest algorithm.Naive Bayes algorithm is the simplest and fastest classification algorithm for a large chunk of data. In various applications such as spam filtering, text classification, sentiment analysis, and recommendation systems, Naive Bayes classifier is used successfully. It uses the Bayes probability theorem for unknown class prediction.The Naive Bayes classification technique is a simple and powerful classification task in machine learning. The use of Bayes’ theorem with a strong independence assumption between the features is the basis for naive Bayes classification.

Simple Bayes or independent Bayes models are other names for Naive Bayes models. All of these terms refer to the classifier's decision rule using Bayes' theorem. In practice, the Bayes theorem is applied by the Naive Bayes classifier. The power of Bayes' theorem is brought to machine learning with this classifier.By using naives the classifier is fitted on the train_sentences and is used to predict labels for the test_sentences. The accuracy of the prediction on the test set comes out to be 75.81967213114754%, which is pretty good. We calculate accuracy using 'accuracy_score' from sklearn.metrics.Another algorithm which we have used is Random Forest algorithm Random Forest is one of the most popular and commonly used algorithms by Data Scientists. 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 for classification and regression tasks. In this tutorial, we will understand the working of random forest and implement random forest on a classification task.the working of the random forest algorithm in machine learning, we must look into the ensemble learning technique. Ensemble simply means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.Ensemble uses two types of methods:

- 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.

- 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, XGBOOST.

In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.

Individual decision trees are constructed for each sample.Each decision tree will generate an output.Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively.After using random forest algorithm we got an accuracy score of 75.99043715846995%.

- The eight step in the implementation is to predict with test data here we will be evaluating the performance of the machine algorithms.Train-Test Split is generally used for classification or regression problems and can be used for any supervised algorithm. We would be taking a data set which is given as the input and divide it into two subsets. The first subset is used to fit the model and is referred to as the training data set. The second subset is not used to train the model, instead the input element of the dataset is provided to the model, then predictions are made and compared to the expected values.The second dataset is referred to as the test dataset.
- The ninth step in the implementation is to evaluate the score of the model.Score of the model or predicted value,can be in many different formats, depending on the model and your input data.For classification models, score model outputs, a predicted value for the class, as well as the probability of the predicted value. For regression models, the score model generates just the predicted numeric value.

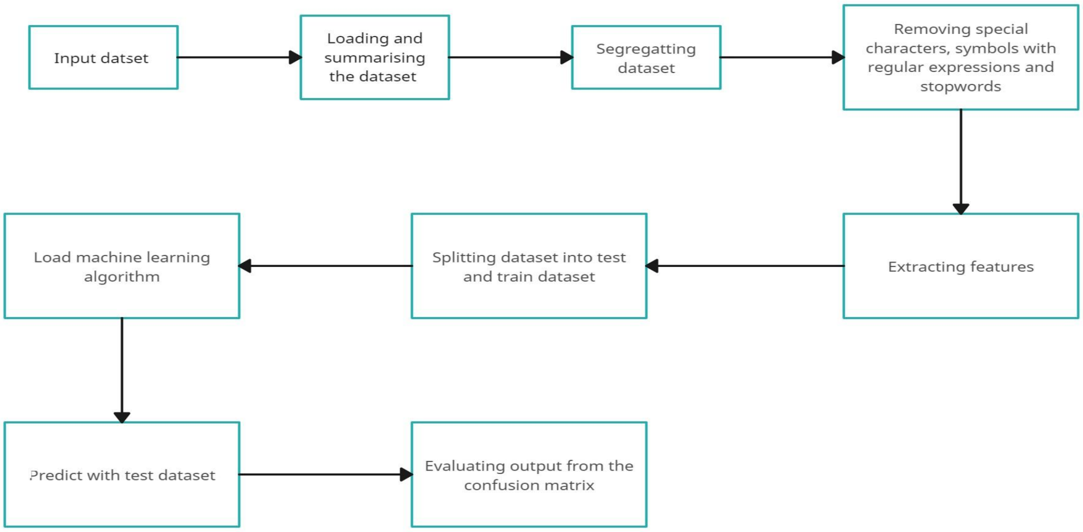

Input datset
Loading and summarising the dataset
Segregatting dataset
Removing special characters, symbols with regular expressions and stopwords
Extracting features
Splitting dataset into test and train dataset
Load machine learning algorithm
Predict with test dataset
Evaluating output from the confusion matrix

Fig 7.1 Implementation steps

7.3 Source Code

We use following libraries in the every python classes to implement the functionality of the project, which is as followed :

```
!pip install nltk
import numpy as np
import pandas as pd
import re #Regular expressions
import nltk
import matplotlib.pyplot as plt

from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from google.colab import files

uploaded = files.upload()
dataset = pd.read_csv('dataset.csv')
print(dataset.shape)
print(dataset.head(5))
features = dataset.iloc[:,10].values
labels = dataset.iloc[:,1].values
print(labels)
processed_features = []
  for sentence in range(0, len(features)):
    processed_feature = re.sub(r'\W',' ', str(features[sentence]))
    processed_feature = re.sub(r'\s+[a-zA-Z]\s+',' ',processed_feature)
    processed_feature = re.sub(r'\^[a-zA-Z]\s+',' ',processed_feature)
    processed_feature = re.sub(r'\s+',' ',processed_feature, flags=re.I)
    processed_feature = re.sub(r'^b\s+',' ',processed_feature)
    processed_feature = processed_feature.lower()
    processed_features.append(processed_feature)
nltk.download('stopwords')
vectorizer = TfidfVectorizer (max_features=2500, min_df=7, max_df=0.8,
stop_words=stopwords.words('english'))
processed_features = vectorizer.fit_transform(processed_features).toarray()
print(processed_features)

X_train, X_test, y_train, y_test = train_test_split(processed_features, labels,
test_size=0.2, random_state=0)
text_classifier = RandomForestClassifier(n_estimators=200, random_state=0)
text_classifier.fit(X_train, y_train)

nb=MultinomialNB()
nb.fit(X_train, y_train)

predictions = text_classifier.predict(X_test)
test_pred = nb.predict(X_test)

rd_score = accuracy_score(y_test,predictions)
nb_score = accuracy_score(y_test, test_pred)
#print(rd_score,nb_score)
if(rd_score> nb_score):
  print("Random forest classifier is the best algorithm for the project with an
accuracy of :", rd_score)
elif (rd_score< nb_score):
  print("Naive` Bayes is the best algorithm for the project with an accuracy of
:",nb_score)
else:
  print("Both have equal accuracy !")
```

```
from sklearn import metrics
import itertools
def plot_confusion_matrix(cm,classes,
                    normalize=False,
                    title='Confusion matrix',
                    cmap=plt.cm.Blues):
  plt.imshow(cm, interpolation='nearest', cmap=cmap)
  plt.title(title)
  plt.colorbar()
  tick_marks = np.arange(len(classes))
  plt.xticks(tick_marks, classes)
  plt.yticks(tick_marks, classes)

  thresh = cm.max() / 2.
  for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    plt.text(j, i, cm[i,j],
            horizontalalignment="center",
            color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

cm = metrics.confusion_matrix(y_test, predictions, labels=['negative', 'neutral',
'positive'])
plot_confusion_matrix(cm,classes=['negative','neutral','positive'])
```

CHAPTER-8

SYSTEM TESTING

The main use of testing is to find out errors. Testing is the way toward attempting to find each possible flaw or shortcoming in a work item. It gives a way to deal with checking the helpfulness of parts, sub-assemblies, social occasions just as a finished thing. It is the path toward working on programming with the point of ensuring that the Software system satisfies its necessities and customer wants and does not bomb in an unacceptable manner. There are various sorts of tests. Each test type keeps an eye on a specific testing need. Testing permits to expel the mistakes and improve the framework execution. There are numerous kinds of tests which enable us to improve our venture execution and to make it mistake free. What's more we likewise have tests which encourage us to check singular modules autonomously and furthermore to check complete framework together according to our convenience.

8.1 Types of Tests

8.1.1 Unit Testing

Unit testing incorporates the arrangement of analyses that favor that within program basis is working properly, and that program information sources produce significant yields. It checks whether little segments are working appropriately or not. Every single decision branch and inside code stream should be endorsed. It is the attempting of individual programming units of the application; it is done after the completion of an individual unit before fuse. This is an auxiliary attempt that relies upon learning of its improvement and is prominent.

Unit tests perform fundamental tests at section level and test a specific business system, application, or possibly structure plan. Unit tests ensure that all of a thoughtful method for a business technique performs unequivocally to the recorded points of interest and contains obviously portrayed data sources and foreseen results. A unit test encourages you to discover which part is broken in your application and fixes it quicker.The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure.

8.1.2 Integration Testing

Integration tests are expected to test joined programming modules to choose whether everything considered continues running as one program. Testing is event driven and is dynamically stressed over the crucial after effect of screens or fields. Combination tests show that in spite of the way that the sections were autonomously satisfied, as showed up by successfully unit testing, the gathering of portions are correct and unsurprising.

Combination testing is expressly a way for revealing the issues that rise up out of the gathering of these portions. Integration testing permits to discover blunders because of unexpected communication between the framework and the sub-framework segments. We test the product in order to test and to identify all the potential mistakes in our undertaking once we complete the source code and before conveying it to the clients.

The techniques for performing tests. These techniques provide guidance for testing. To test the internal logic of the software components. To test the input and output domains of a program and to uncover the errors in program function, behavior and performance. We can test the software by using two methods:

- White Box testing: In this the internal logic program is being checked by using different test case design techniques.
- Black Box testing: In this the software requirements are tested by using different test case design techniques. Both the above mentioned techniques help in finding out the maximum number of errors with minimal time and effort.

The lowest level components are tested first in bottom-up testing. They are then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested. All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.

8.1.3 Performance Testing

This test is done to find the run-time performance of the software with the context of the integrated system. These tests can be carried out throughout the testing process. For example, the performance of individual modules are accessed during white box testing under unit testing. However, performance testing is frequently not performed against a specification; e.g., no one will have expressed what the maximum acceptable response time for a given population of users should be.

Performance testing is frequently used as part of the process of performance profile tuning. The idea is to identify the "weakest link" – there is inevitably a part of the system which, if it is made to respond faster, will result in the overall system running faster. It is sometimes a difficult task to identify which part of the system represents this critical path, and some test tools include (or can have add-ons that provide) instrumentation that runs on the server (agents) and reports transaction times, database access times, network overhead, and other server monitors, which can be analyzed together with the raw performance statistics. It is always helpful to have a statement of the likely peak number of users that might be expected to use the system at peak times. If there can also be a statement of what constitutes the maximum allowable 95 percentile response time, then an injector configuration could be used to test whether the proposed system met that specification.

8.1.4 Verification and Validation

Testing procedure is a piece of subject alluding to checking and approval of our task. We have to find the framework determinations and we should attempt to meet the details of the client and to fulfill the client, for this reason, we need to check and approve the item and we have to ensure that everything is working appropriately. Check and approval are the two unique things. One is performed to guarantee that the product is working accurately and to implement a particular usefulness and the other is done to guarantee if the client prerequisites are appropriately met or not by the finished result. Check is progressively similar to 'would we say we are building the item right and approval is increasingly similar to 'would we say we are building the correct item.

8.2 Testing Table

PAGE	TEST CASE	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
Importing required libraries	To complete the project we require several python, natural language processing and scikit learn libraries to perform the operations	Using "! pip" we install all the required libraries for the demonstration of the project	The libraries after installation displays the message that it is successfully installed.	As expected	Pass
Uploading dataset	To analyze the sentiment we upload a dataset of feedback or opinions from the users across various social mediums like twitter, amazon and flipkart.	Dataset would be successfully uploaded for the proper sentiment analysis.	The dataset needs to be successfully uploaded.	As expected	Pass
Dataset size limit	Uploaded dataset can be of any countable data.	Dataset would be accepted whatever be the size of the data set.	Uploading dataset of size 14000+ items are successfully updated for the sentiment analysis.	As expected	Pass
Segregation of input and output	The dataset is uploaded and data should be segregated for both input and output operations.	The dataset should segregate into both input and output for analysis.	The user will be notified with the input data displayed on the screen which is going to be trained.	As expected	Pass

Data cleaning	The uploaded dataset should be cleaned from raw characters, spaces and null values.	The dataset should be cleaned from raw data using regular expressions.	The user will be displayed with the cleaned data ready for the analysis.	As expected	Pass
Feature extraction	The features like stop words are extracted from the input dataset.	The dataset should be operated to obtain features using NLTK libraries.	The user will be displayed with the data after extracting the features.	As expected	Pass
Splitting the dataset	Divide the dataset into test and train categories for analysis.	The dataset would split into train and test data for operations	The user is displayed with the amount of dataset split for training dataset	As expected	Pass
Loading machine learning classification algorithms	To classify the sentiment in the dataset, we use classification algorithms.	The algorithms are loaded through sklearn library	We can see that random forest and naive bayes algorithm is loaded through sklearn.	As expected	Pass
Estimating the best algorithm	The efficiency produced by the algorithm is estimated.	The algorithms are tested for best efficiency	We can see that random classifier is has best efficiency	As expected	Pass
Generating output response	The output for the sentiment analysis is generated.	The output as confusion matrix	The sentiment analysis is done and generates output	As expected	Pass

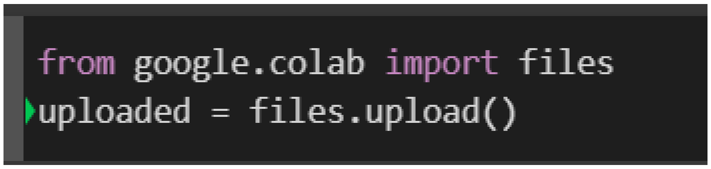
CHAPTER- 9

RESULTS AND OUTPUT SCREENS

9.1 Evaluating Results

This is the confusion matrix of the user given input data set. It will collect the dataset next it will load and summarize the dataset then segregates dataset into X & Y and removes special characters, symbols with regular expressions, stop words and extracts features from the dataset then splits data into train and test then loads and applies machine learning algorithms such as Naive Bayes and Random Forest algorithms lastly predicts with which algorithm we are having best accuracy for the given dataset and evaluate the result of dataset with best algorithm as below with confusion matrix.

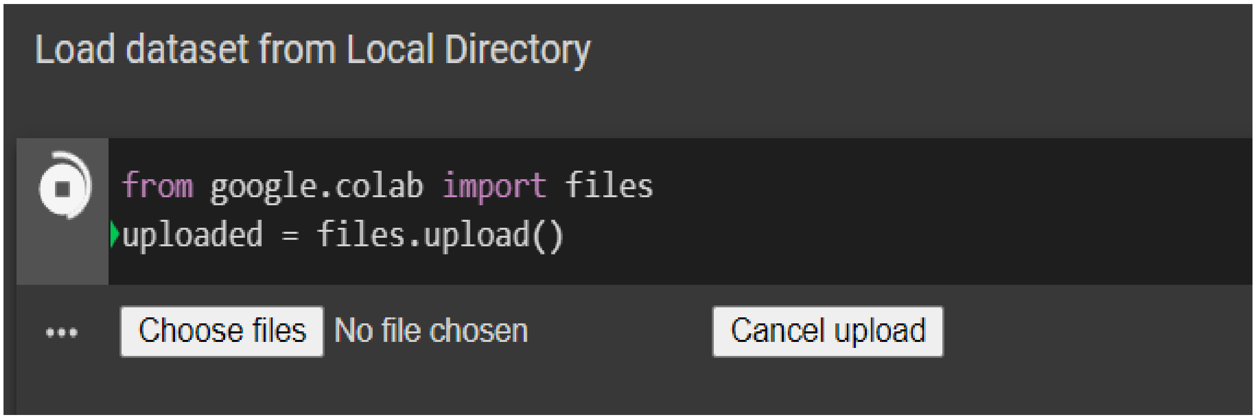
Input: We need to execute this block of code to upload the input file for processing data.



```
from google.colab import files
uploaded = files.upload()
```

Fig 9.1 Process to upload dataset

After execution of the above block of code we will be prompted to upload the input file from the local directory of our system or to cancel upload.

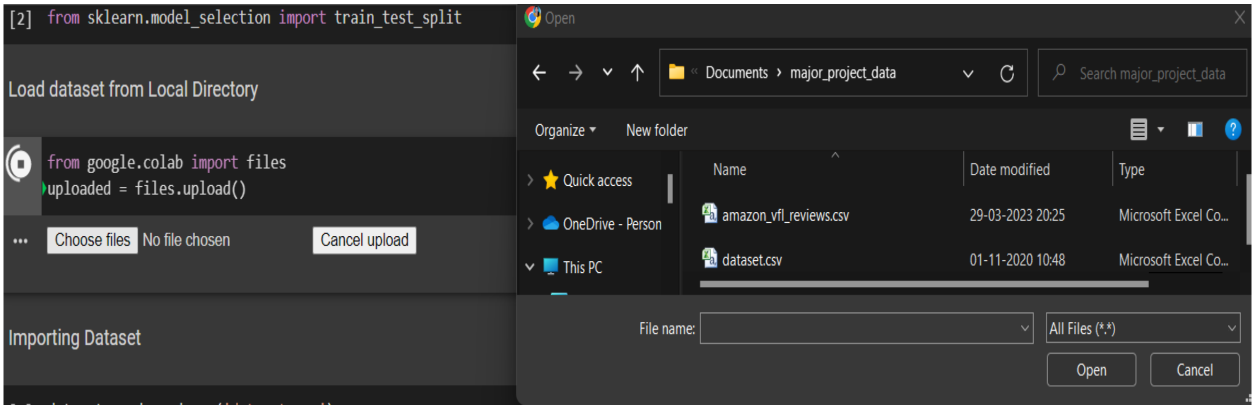


Load dataset from Local Directory

```
from google.colab import files
uploaded = files.upload()
```

Fig 9.2 Loading dataset from local directory

If we choose the option to choose files then we can upload our input file from our local directory or if we want to cancel the upload we can cancel it anytime by clicking cancel upload.



```
[2] from sklearn.model_selection import train_test_split
```

Load dataset from Local Directory

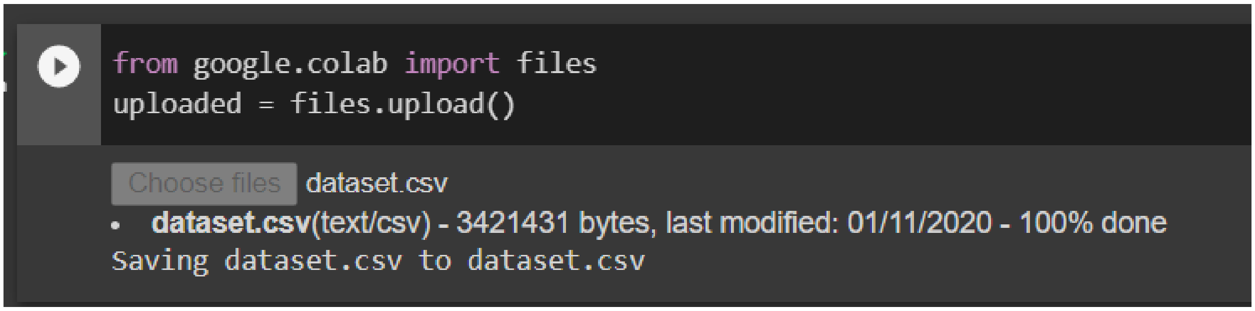
```
from google.colab import files
uploaded = files.upload()
```

Choose files No file chosen Cancel upload

Importing Dataset

Fig 9.3 Choosing dataset

After we have chosen the input file from the directory it will be uploaded and we can see a message saying that file is uploaded.



```
from google.colab import files
uploaded = files.upload()
```

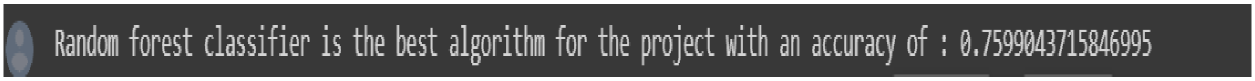
Choose files dataset.csv

- **dataset.csv**(text/csv) - 3421431 bytes, last modified: 01/11/2020 - 100% done

```
Saving dataset.csv to dataset.csv
```

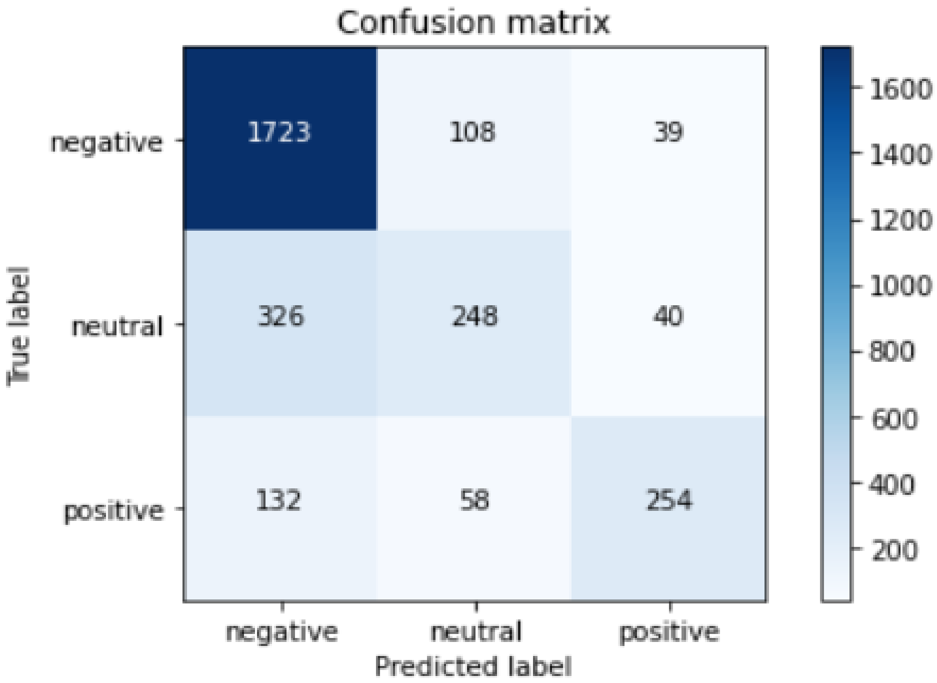
Fig 9.4 Result after uploading dataset

Output : After going through all the processing steps of the algorithm we will get the accuracy of the best algorithm and sentiment behind the input file will be generated in the form of a confusion matrix.



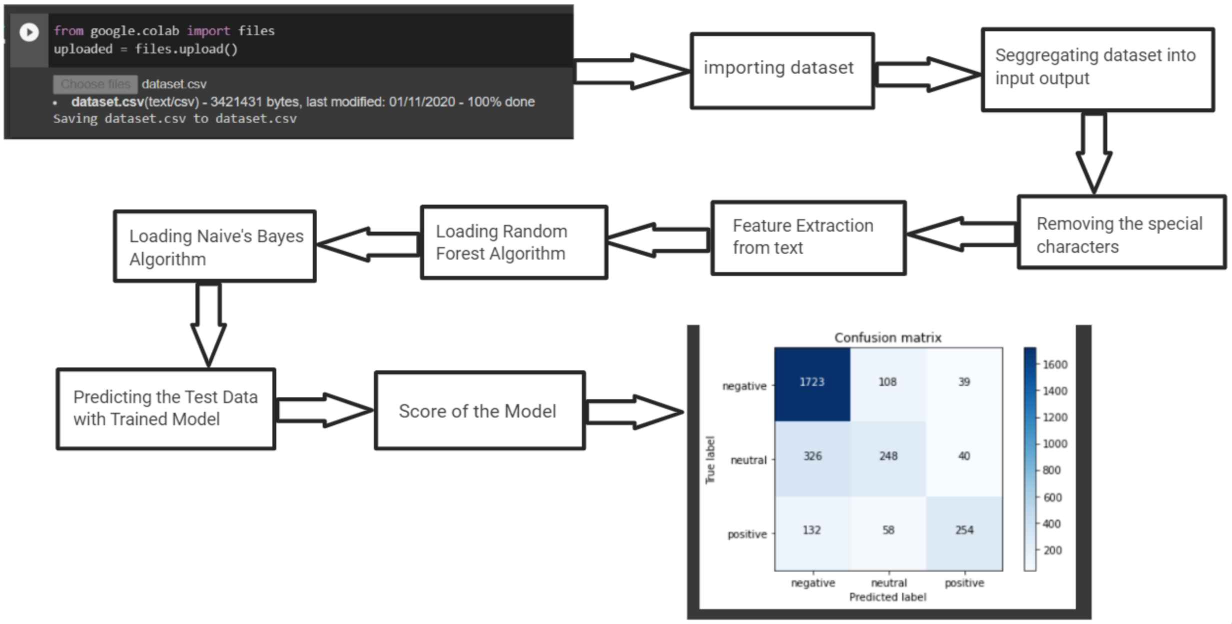
```
Random forest classifier is the best algorithm for the project with an accuracy of : 0.7599043715846995
```

Fig 9.5 Best algorithm with accuracy



	negative	neutral	positive
negative	1723	108	39
neutral	326	248	40
positive	132	58	254

Fig 9.6 Confusion matrix of input data set



```
from google.colab import files
uploaded = files.upload()
```

Choose files dataset.csv

- **dataset.csv**(text/csv) - 3421431 bytes, last modified: 01/11/2020 - 100% done

```
Saving dataset.csv to dataset.csv
```

importing dataset
Seggregating dataset into input output
Removing the special characters
Feature Extraction from text
Loading Random Forest Algorithm
Loading Naive's Bayes Algorithm
Predicting the Test Data with Trained Model
Score of the Model
Confusion matrix
negative
neutral
positive
True label
Predicted label
1723
108
39
326
248
40
132
58
254
1600
1400
1200
1000
800
600
400
200

Fig 9.7 Project flow

CHAPTER- 10

CONCLUSION AND FUTURE ENHANCEMENT

In the current review based system the customers are reviewing the products using the rating based system which is not useful to exactly know the review of the customer like where the product failed or product is good.

In the proposed application we can get to know what the feedback is whether a positive or negative and on what basis we have decided it is positive or negative and we can also know the time of application to review product and number of words present in the sentences which is given as input to the application system.With help of machine learning algorithms and natural language processing it is so easy and efficient to get feedback of customer.

The sentiment analysis is developed to analyze the customer perception to an entity in the race of success in the day-to-day market. The project using machine learning and convolution NLP libraries has increased the efficiency of the analysis. For further development we can also use this approach in analyzing the emotions from images, voice and files. Rather than giving positive, negative and neutral as an output, as per our project it is possible to understand the sentiment analysis through much more dimensions using a confusion matrix.

CHAPTER- 11
REFERENCES

[1] https://research.google.com/colaboratory/faq.html

[2] https://numpy.org/doc/stable/user/absolute_beginners.html

[3]https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/

[4] https://www.turing.com/kb/an-introduction-to-naive-bayes-algorithm-for-beginners

[5] https://www.turing.com/kb/random-forest-algorithm

[6] https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications

[7] Brian Keith Norambuena* , Exequiel Fuentes Lettura and Claudio Meneses Villegas: "Sentiment analysis and opinion mining", DOI 10.3233/IDA-173807, 2019. https://www.researchgate.net/publication/331291125.

[8] Xing Fang and Justin Zhan: "Sentiment analysis using product review data", Fang and Zhan Journal of Big Data 2:5 DOI 10.1186/s40537-015-0015-2, 2015.

[9] Zulfadzli Drus, Haliyana Khalid: " Sentiment Analysis in Social Media and Its Application: Systematic Literature Review ", Procedia Computer Science 16 707–714 Science direct, Elsevier, 2019.

[10] Tetsuya Nasukawa, Jeonghee Yi. Sentiment analysis: "Capturing favorability using natural language processing ", https://doi.org/10.1145/945645.945658,2003.

[11] Aliza Sarlan, Chayanit Naam, Shuid Basri : "Twitter sentiment analysis", ISSN No:-2456-2165,2019.

[12] Rohit Raj Sehgal, Shubham agarwal, Gaurav Raj: "Interactive voice response using sentiment analysis in automatic speech recognition systems", (ICACCE-2018) Paris, 2019.