DRUG RECOMMENDATION SYSTEM BASED ON SENTIMENT ANALYSIS OF DRUG REVIEWS

Using Machine Learning

SUBMITTED TO

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In the partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

CH K N M DATTATREYA	**19NG1A0568**
N SAI SEKHAR	**19NG1A05A0**
V AKANKSHA	**19NG1A05B9**
R RAMYA SRI	**19NG1A05A5**

Under the Esteemed Guidance of

B V PRAVEEN KUMAR M.tech

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

USHARAMA

COLLEGE OF ENGINEERING AND TECHNOLOGY

AUTONOMOUS

(Approved by AICTE and JNTUK, Kakinada)

(ON NH 16, TELAPROLU, NEAR GANNAVARAM - 521109)

2019-2023

USHA RAMA
COLLEGE OF ENGINEERING AND TECHNOLOGY

AUTONOMOUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

USHA RAMA COLLEGE OF ENGINEERING AND TECHNOLOGY
(Affiliated of to JNTU Kakinada, Approved by A.I.C.T.E, New Delhi)
TELAPROLU, UNGUTURU MANDAL, KRISHNA DISTRICT-521109
2019-2023

CERTIFICATE

This is to certify that this project entitled **"DRUG RECOMMENDATION SYSTEM BASED ON SENTIMENT ANALYSIS OF DRUG REVIEWS USING MACHINE LEARNING"** is the bonafide work of **CH K N M DATTATREYA(19NG1A0568),N SAI SEKHAR(19NG1A05A0),V AKANKSHA(19NG1A05B9),R RAMYA SRI(19NG1A05A5)** who carried out the work under our supervision, and submitted in partial fulfillment of the requirements for the award of the degree in Bachelor of Technology in Computer Science & Engineering, during the academic year 2022-23.

Project Guide
B V PRAVEEN KUMAR M.tech
Assistant Professor

Head of the Department
Dr. S M ROY CHOUDRI Ph.D
Professor

Signature of External Examiner

DECLARATION

We hereby declare that the project entitled **"DRUG RECOMMENDATION SYSTEM BASED ON SENTIMENT ANALYSIS OF DRUG REVIEWS USING MACHINE LEARNING"** is the work done by us during the academic year 2022-2023 and is submitted inpartial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA**.

BY

CH. K N M DATTATREYA	**(19NG1A0568)**
N SAI SEKHAR	**(19NG1A05A0)**
V.AKANKSHA	**(19NG1A05B9)**
R.RAMYA SRI	**(19NG1A05A5)**

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to our Honorable Chairman **SRI. S. RAMABRAHMAM** for the guidance and advice which is given and for providing sufficient resources.

We are extremely thankful to **Dr. K RAJASEKHARA RAO**, Director of USHA RAMA COLLEGE OF ENGINEERING AND TECHNOLOGY, TELAPROLU for giving a golden opportunity to our education and project work.

We wish to avail this opportunity to express our gratitude to **Dr. G V K S V PRASAD**, Principal, URCE for his continuous support and giving valuable suggestions during the entireperiod of the project work.

We take this opportunity to express our gratitude to **Dr. S M ROY CHOUDRI**, Head of the Department and for his valuable support and motivation at each and every point in successful completion of the project.

We would like to express our gratitude to **Mr B V PRAVEEN KUMAR ,**Assistant Professor in the Department of Computer Science and Engineering,our project guide for motivating us to make our project successful. We grateful for his precious guidance and suggestions

We also place our floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the project.

BY

CH. K N M DATTATREYA (19NG1A0568)

N SAI SEKHAR (19NG1A05A0)

V.AKANKSHA (19NG1A05B9)

R.RAMYA SRI (19NG1A05A5)

DRUG RECOMMENDATION SYSTEM BASED ON SENTIMENT ANALYSIS OF DRUG REVIEWS USING MACHINE LEARNING

ABSTRACT

ABSTRACT

Since corona virus has shown up, inaccessibility of legitimate clinical resources is at its peak,

Like the shortage of specialists and healthcare workers, lack of proper equipment and medicines.

The entire medical fraternity is in distress, which results in numerous individual's demise.Due to unavailability, individuals started taking medication independently without appropriate consultation,making the health condition worse than usual. As of late, machine learning has been valuable in numerous applications, and there is an increase and innovative work for automation. This paper intends to present a drug recommender system that can drastically reduce specialists heap.

In this research, we build a medicine recommendation system that uses patient reviews to predict the sentiment using various vectorization processes like Bow, TF IDF,Word2Vec,and Manual Feature Analysis, which can help recommend the top drug for a given disease by different classification algorithms.The predicted sentiments were evaluated by precision, recall, f1score,accuracy, and AUC score. The results show that classifier Linear SVC using TF IDF vectorization Outperforms all other models with 93% accuracy.

TABLE OF CONTENTS

TOPIC	PAGE NO
1. INTRODUCTION	01
1.1 Literature Survey	03
1.1.1. Machine Learning	04
1.1.2. Features of Machine Learning	08
1.1.3. Existing System	10
1.1.4. Proposed System	11
2. AIM & SCOPE	12
2.1. Feasibility study	13
2.1.1. Technical Feasibility	13
2.1.2. Operational Feasibility	14
2.1.3. Economic Feasibility	14
2.2. System Requirements Specification	15
2.2.1. Functional Requirements	15
2.2.2. Non-Functional Requirements	16
2.2.3. Software Requirements	17
2.2.4. Hardware Requirements	17
3. CONCEPTS & METHODS	18
3.1. Problem Definition	19
3.2. Proposed Description	20
3.2.1. Algorithms Proposed	21
3.2.2. Modules	22
3.3. System analysis methods	23
3.3.1. Use case Diagram	23
3.3.2. Dataflow Diagram	24
3.4. System design	25
3.4.1. System Architecture	25
3.4.2. Class Diagram	26
3.4.3. Sequence Diagram	27
4. IMPLEMENTATION	28

4.1. Tools used 29
4.2. Pseudo code 32
4.3. Component Diagram 41
4.4. Deployment Diagram 42
5. SCREEN SHOTS 43
6. TESTING 48
7. SUMMARY & CONCLUSION 53
8. FUTURE ENHANCEMENTS 55
9. BIBILOGRAPHY 57
10. RESEARCH PAPER

CHAPTER - 1

INTRODUCTION

1. INTRODUCTION

With the number of corona virus cases growing exponentially, the nations are facing a shortage of doctors, particularly in rural areas where the quantity of specialists is less compared to urban areas. A doctor takes roughly 6 to 12 years to procure the necessary qualifications. Thus, the number of doctors can't be expanded quickly in a short time frame. A Medicine framework ought to be energized as far as possible in this difficult time [1]. Clinical blunders are very regular nowadays. Over 200 thousand individuals in China and 100 thousand in the USA are affected every year because of prescription mistakes.

Over 40% medicine, specialists make mistakes while prescribing since specialists compose the solution as referenced by their knowledge, which is very restricted [2][3]. Choosing the top level medication is significant for patients who need specialists that know wide-based information about microscopic organisms, antibacterial medications, and patients [6]. Every day a new study comes up with accompanying more drugs, tests, accessible for clinical staff every day. Accordingly, it turns out to be progressively challenging for doctors to choose which treatment or medications to give to a patient based on indications, past clinical history.

A medication recommend er framework is truly vital with the goal that it can assist specialists and help patients to build their knowledge of drugs on specific health conditions. A recommend er framework is a customary system that proposes an item to the user, dependent on their advantage and necessity. These frameworks employ the customers' surveys to break down their sentiment and suggest a recommendation for their exact need. In the drug recommend er system, medicine is offered on a specific condition dependent on patient reviews using sentiment analysis and feature engineering. Sentiment analysis is a progression of strategies, methods, and tools for distinguishing and extracting emotional data, such as opinion and attitudes, from language [7].

On the other hand, Featuring engineering is the process of making more features from the existing ones; it improves the performance of models. This examination work separated into five segments: Introduction area which provides a short insight concerning the need of this research, Related works segment gives a concise insight regarding the previous examinations on this area of study, Methodology part includes the methods adopted in this research, The Result segment evaluates applied model results using various metrics, the Discussion section contains limitations of the framework, and lastly, the conclusion section.

1.1. LITERATURE SURVEY

With a sharp increment in AI advancement, there has been an exertion in applying machine learning and deep learning strategies to recommend frameworks.These days,recommend frameworks are very regular in the travel industry, e-commerce, restaurant, and so forth. Unfortunately, there are a limited number of studies available in the field of drug proposal framework utilizing sentiment analysis on the grounds that the medication reviews are substantially more intricate to analyze as it incorporates clinical wordings like infection names, reactions, a synthetic names that used in the production of the drug [8].

The study [9] presents Galen OWL, a semantic-empowered online framework, to help specialists discover details on the medications. The paper depicts a framework that suggests drugs for a patient based on the patient's infection, sensitivities, and drug interactions. For empowering Galen OWL, clinical data and terminology first converted to ontological terms utilizing worldwide standards, such as ICD-10 and UNII, and then correctly combined with the clinical information.

Leila Sun [10] examined large scale treatment records to locate the best treatment prescription for patients. The idea was to use an efficient semantic clustering algorithm estimating the similarities between treatment records. Likewise, the author created a framework to assess the adequacy of the suggested treatment. This structure can prescribe the best treatment regimens to new patients as per their demographic locations and medical complications. An Electronic Medical Record (EMR) of patients gathered from numerous clinics for testing. The result shows that this framework improves the cure rate.In this research [11], multilingual sentiment analysis was performed using Naive Bayes and Recurrent Neural Network (RNN). Google translator API was used to convert multilingual tweets into the English language. The results exhibit that RNN with 95.34% outperformed Naive Bayes, 77.21%..

The study [12] is based on the fact that the recommended drug should depend upon the patient's capacity. For example, if the patient's immunity is low, at that point, reliable medicines ought to be recommended. Proposed a risk level classification method to identify the patient's immunity. For example, in excess of 60 risk factors, hypertension, liquor addiction, and so forth have been adopted, which decide the patient's capacity to shield himself from infection. A web-based prototype system was also created, which uses a decision support system that helps doctors select first-line drugs.

Xiaohong Jiang et al. [13] examined three distinct algorithms, decision tree algorithm, support vector machine (SVM), and back propagation neural network on treatment data. SVM was picked for the medication proposal module as it performed truly well in each of the three unique boundaries - model exactness, model proficiency, model versatility. Additionally, proposed the mistake check system to ensure analysis, precision and administration quality.

Mohammad Mehedi Hassan et al. [14] developed a cloud- assisted drug proposal (CADRE). Considering the significance of hashtags in sentiment analysis, Jiugang Li et al. [15] constructed a hashtag recommend er framework that utilizes the skip-gram model and applied conventional neural networks (CNN) to learn semantic sentence vectors. These vectors use the features to classify hashtags using LSTM RNN.

1.1.1. MACHINE LEARNING

Tom Mitchell states machine learning as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience”. Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbour Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.,

The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter sub field focuses more on exploratory data analysis and is known as unsupervised learning.

With in the field of data analytic s, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytic s. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

Machine learning implementations are classified into three major categories, depending on the nature of the learning “signal” or “response” available to a learning system which are as follows:

Supervised learning: When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.

Unsupervised learning: When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms. As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of

similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

Reinforcement learning: When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error. Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others.

In this case, an application presents the algorithm with examples of specific situations, such as having the gamer stuck in a maze while avoiding an enemy. The application lets the algorithm know the outcome of actions it takes, and learning occurs while trying to avoid what it discovers to be dangerous and to pursue survival. You can have a look at how the company Google Deep Mind has created a reinforcement learning program that plays old Atari's video 3 games. When watching the video, notice how the program is initially clumsy and unskilled but steadily improves with training until it becomes a champion.

Semi-supervised learning:

Where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing. Supervised Learning the majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data. It is called supervised learning because the process of an algorithm learning from the training data set can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm interactively makes predictions on the training data and is corrected by the teacher. Learning stopswhen the algorithm achieves an acceptable level of performance.

Types of Supervised Learning:

Classification: It is a Supervised Learning task where output is having defined labels (discrete value). For example, in above Figure A, Output – Purchased has defined labels i.e., 0 or 1; 1 means the customer will purchase and 0 means that customer won't purchase. The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy. It can be either binary or multi class classification. In binary classification, model predicts either 0 or 1; yes or no but in case of multi class classification, model predicts more than one class. Example: G mail classifies mails in more than one classes like social, promotions, updates, forum.

Regression: It is a Supervised Learning task where output is having continuous value. Example in above Figure B, Output – Wind Speed is not having any discrete value but is continuous in the particular range. The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value. The smaller the error the greater the accuracy of our regression model.

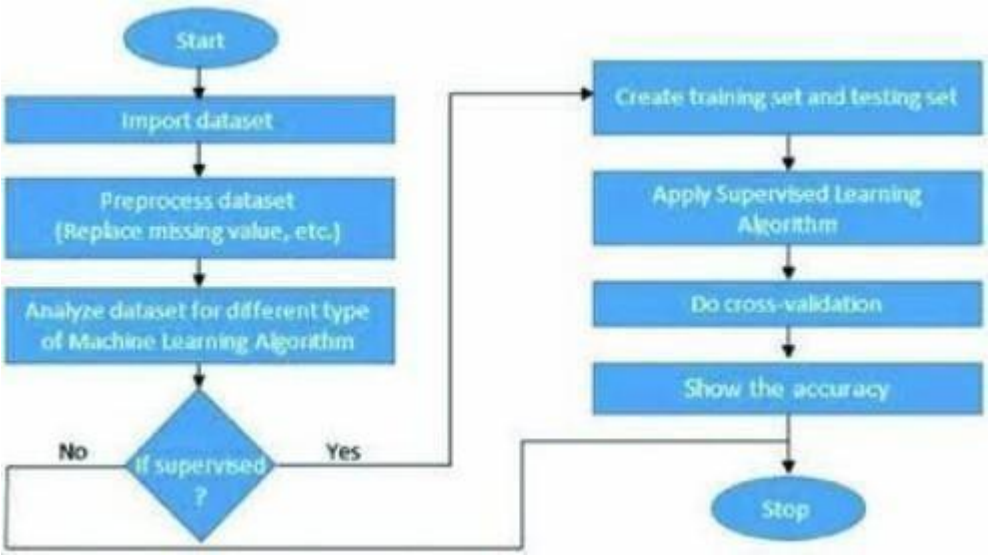


Fig 1.1.1. FLOW CHART OF SUPERVISED LEARNING ALGORITHM

Classification:

Data mining is the process of extracting knowledge-able information from huge amounts of data. It is an integration of multiple disciplines such as statistics, machine learning, neural networks and pattern recognition. Data mining extracts biomedical and health care knowledge for clinical decision making and generates scientific hypotheses from large medical data.

Association rule mining and classification are two major techniques of data mining. Association rule mining is an unsupervised learning method for discovering interesting patterns and their association in large data bases.

Classification is a supervised learning method used to find class labels for unknown samples. Classification is the task of assigning an object's tone of special predefined categories. It is pervasive problem that encompasses many applications.

Classification is designed as the task of learning a target function F that maps each attribute set A to one of the predefined class labels C. The target function is also known as classification model. A classification model is useful for mainly two purposes.

1) descriptive modelling.
2) Predictive modelling.

Classification is the process of recognizing, understanding, and grouping ideas and objects into categories or "sub-populations." Using pre categorized training datasets, machine learning programs use a variety of algorithms to classify future datasets into categories.

Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. One of the most common uses of classification is filtering emails into "spam" or "non-spam."

In short, classification is a form of "pattern recognition," with classification algorithms applied to the training data to find the same pattern (similar words or sentiments, number sequences, etc.) in future sets of data.

Classification can be performed on structured or unstructured data. Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Few of the terminologies encountered in machine learning – classification:

Classifier: An algorithm that maps the input data to a specific category.

Classification model: A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.

Feature: A feature is an individual measurable property of a phenomenon being observed. Binary

Classification: Classification task with two possible outcomes. E.g., Gender classification (Male / Female).

Multi-class classification: Classification with more than two classes. In multi class classification each sample is assigned to one and only one target label. E.g., An animal can be cat or dog but not both at the same time.

Multi-label classification: Classification task where each sample is mapped to a set of target labels (more than one class). E.g., A news article can be about sports, a person, and location at the same time.

Applications of Classification Algorithms:

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumour cells identification.
- Sentiment analysis
- Drug's classification
- Facial key points detection
- Pedestrians' detection in an automotive car driving.

1.1.2. FEATURES OF MACHINE LEARNING

- It is nothing but automating the Automation.
- Getting computers to program themselves.
- Writing Software is bottleneck.
- Machine leaning models involves machines learning from data without the help of humans orany kind of human intervention.
- Machine Learning is the science of making of making the computers learn and act like humans byfeeding data and information without being explicitly programmed.
- Machine Learning is totally different from traditionally programming, here data and output is given to the computer and in return it gives us the program which provides solution to the various problems. Below is the figure.

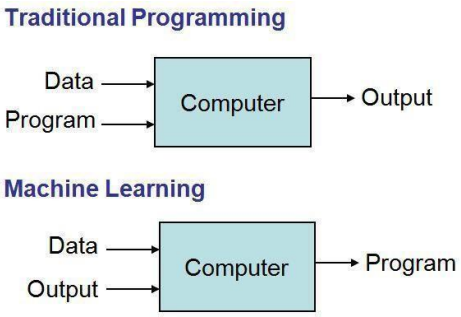
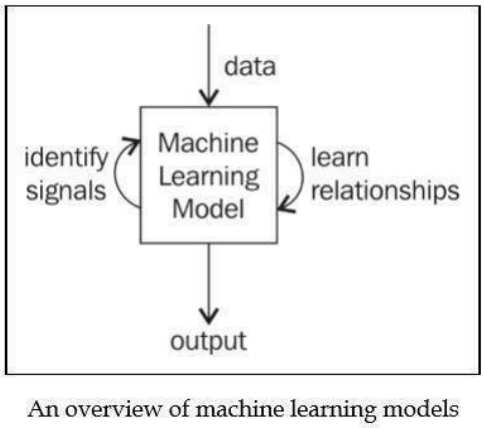


Fig 1.1.2 Traditional Programming vs Machine Learning

- Machine Learning is a combination of Algorithms, Datasets,and Programs.
- There are Many Algorithms in Machine Learning through whichwe will provide us the exact solutionin predicting the disease of the patients.
- How Does Machine Learning Works?
- Solution to the above question is Machine learning works by taking in data, finding relationships within that data and then giving the output.



An overview of machine learning models

Fig 1.1.3 Machine Learning Model

There are various applications in which machine learning is implemented such as Web search, computingbiology, finance, e-commerce, space exploration, robotics, social networks, debugging and much more.

1.1.3. EXISTING SYSTEM

The Existing system is based on the fact that the recommended drug should depend upon the patient's capacity. For example, if the patient's immunity is low, at that point, reliable medicines ought to be recommended. Proposed a risk level classification method to identify the patient's immunity. For example, in excess of 60 risk factors, hypertension, liquor addiction, and so forth have been adopted, which decide the patient's capacity to shield himself from infection. A web-based prototype system was also created, which uses a decision support system that helps doctors select first-line drugs.

DISADVANTAGES:

- Accuracy is Low.
- Computationally costly, cold start, and information sparsity.

1.1.4. PROPOSED SYSTEM

Distinct machine-learning classification algorithms were used to build a classifier to predict the sentiment. After assessing the metrics, all four best-predicted results were picked and joined together to produce the combined prediction. The merged results were then multiplied with normalized useful count to generate an overall score of drug of a particular condition. The higher the score, the better is the drug. The purpose behind is that the more medications individuals search for, the more individuals read the survey regardless of their review is positive or negative, which makes the useful count high.

ADVANTAGES:

- Accuracy is Very high.
- We selected machine learning classification algorithms only that reduces the training time and give faster predictions.

CHAPTER-2

AIM & SCOPE

2. AIM & SCOPE

The main of this research paper is to develop Drug Recommendation. The system provides a platform where Drugs are Recommended to users and their related cure by just providing the symptoms as input The Scope of this project is this can be used for a Remote areas.

2.1. FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase. During system analysis the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

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

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

2.1.1. Technical Feasibility:

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. Thus, it provides an easy access to the users. Therefore, it provides the technical guarantee of accuracy, reliability and security. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

2.1.2. Operational Feasibility:

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is their sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

2.1.3. Economic feasibility:

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies.

2.2. SYSTEM REQUIREMENT SPECIFICATION

A Software Requirements Specification (SRS) – a requirements specification for a software system– is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification is a structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions.

2.2.1. FUNCTIONAL REQUIREMENTS:

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements(also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall character such as costant reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

- Functional Requirementsconcernswiththespecificfunctionsdeliveredbythesystem.
- So,functionalrequirementsarestatementsoftheservicesthatthesystemmustprovide.
- The functional requirements of the system should be both complete and consistent
- Completeness means that all the services required by the user should be defined.
- Consistencymeansthatrequirementsshouldnothaveanycontradictorydefinitions.
- The requirements are usually described in a fairly abstract way. However, functional system requirements describe the system function in details, its inputs and outputs, exceptions and soon.
- Takeuseridandpasswordmatchitwithcorrespondingfileentries. If a match is found then continue else raise an error message.

2.2.2. NON-FUNCTIONAL REQUIREMENTS:

- Non-functional Requirements refer to the constraints or restrictions on the system. They may relate to emergent system properties such as reliability, response time and store occupancy or the selection of language, platform, implementation techniques and tools.
- The non-functional requirements can be built on the basis of needs of the user, budget constraints, organization policies and etc.

1. **Performance requirement:** All data entered shall be up to mark and no flaws shall be there for the performance to be 100%.

2. **Platform constraints:** The main target is to generate an intelligent system to predict the adult height.

3. **Accuracy and Precision:** Requirements are accuracy and precision of the data.

4. **Modifiablity:** Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort(person-months).

5. **Portability:** Since mobile phone is handy so it is portable and can be carried and used when ever required.

6. **Reliability:** Requirements about how often the software fails. The definition of a failuremust be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how toprotect from failure,a strategy for error Prediction, and a strategy for correction.

7. **Security:** One or more requirements about protection of your system and its data.

8. **Usability:** Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people who have registered with the registration page can access their data with the help of login. User interface is simple and efficient and easy touche.

MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to include new functionalities can be added in the project based on the user requirements justly adding the appropriate files to existing project using .net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes inthe project.

SCALABILITY:

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

VALIDATION:

It is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility ofsoftware testers as part of the software development life cycle. Software validation checks thatthe software product satisfies or fits the intended use (high-level checking), i.e., the software meetsthe user requirements, not as specification artefact or as needs of those who will operate the software only;but,as the need so fall the stakeholders.

2.2.3. HARDWARE REQUIREMENTS

System	:Intel i5 (Quadcore Processor isrecommended)
RAM	: Minimum 4GB
Hard Disk	: 250 GB HDD or SSD (SSD Recommended)
Key Board	: Standard Keyboard
Monitor	:15 VGA Colour

2.2.4. SOFTWARE REQUIREMENTS

Operating System	: Windows 8.1 and greater (Windows 10 Recommended)
Platform	: Anaconda Prompt (anaconda3)
FrontEnd	: Html & CSS
Back End	: Python & Datasets
Programming Language	: Python

CHAPTER - 3

CONCEPTS & METHODS

3. CONCEPTS & METHODS

3.1. PROBLEM DEFINITION

Since the arrival of advanced computing, the doctors' still requires the technology in various possible ways like surgical representation process and x-ray photography, but the technology perceptually stayed behind. The method still requires the doctor's information and experience due to alternative factors starting from medical records to weather conditions, atmosphere, blood pressure and numerous alternative factors. The huge numbers of variables are consider as entire variables that are required to understand the complete working process itself, however no model has analyzed successfully. To tackle this drawback, Medical decision support systems must be used. This system is able to assist the doctors to make the correct decision. Medical decision support system refers to both the process of attempting to determine or identify possible diseases or disorder and the opinion reached by this process.

3.2. PROJECT DESCRIPTION

Medical decision could be extremely specialized and difficult job due to alternative factors or in case of rare diseases. The alternative factors include stress; tired misdiagnosis might vary from ignorance of doctors and incomplete information. Standard algorithm may go through the entire variables like prevailing conditions history of medical records, history of family records and various factors relating to the patient records, sheer magnitude of obtainable hidden factors. Differential diagnosis methods can be used to identify the presence of an entity where multiple alternatives are possible and also refers to include the candidate alternatives. This method is needs a process of elimination or obtaining information that shrinks the probability of candidate conditions to negligible levels.

These frameworks employ the customers' surveys to break down their sentiment and suggest a recommendation for their exact need. In the drug recommender system, medicine is offered on a specific condition dependent on patient reviews using sentiment analysis and feature engineering. Sentiment analysis is a progression of strategies, methods, and tools for distinguishing and extracting emotional data, such as opinion and attitudes, from language

3.2.1. ALGORITHM PROPOSED

Bi directional LSTM

Bi-directional LSTM (Long Short-Term Memory) algorithm is a variant of the standard LSTM algorithm used in deep learning for sequential data processing. Bi-directional LSTMs are capable of processing sequential data in both forward and backward directions simultaneously, making them highly effective for tasks that involve analyzing a sequence of data points, such as natural language processing, speech recognition, and time-series analysis.

The bi-directional LSTM algorithm works by utilizing two separate LSTM layers - one that processes the input sequence in the forward direction, and another that processes it in the backward direction. The output of each of these LSTM layers is concatenated, resulting in a final output that takes into account both the past and future context of each data point in the sequence.

One of the key advantages of the bi-directional LSTM algorithm is its ability to capture long-term dependencies in sequential data. This is achieved through the use of memory cells and gates, which allow the algorithm to selectively store and retrieve information from previous time steps as needed.

Another advantage of the bi-directional LSTM algorithm is its ability to handle variable-length input sequences. This is accomplished by dynamically adjusting the size of the hidden state and memory cell vectors in response to the input sequence.

Overall, the bi-directional LSTM algorithm is a powerful tool for processing sequential data, and has been used successfully in a wide range of applications, including natural language processing, speech recognition, and time-series analysis**.**

3.2.2. MODULES

1. Login
2. Patient's Review
3. Recommendation Report
4. Python File Machine Learning Implemented

3.2.2.1. Login(log.html):

This "log.html" page with CCS applied used to take the "Login Credentials" from Doctor or Medic Team in order to login into the "Registration page" for patients to fill their details.

.

3.2.2.2. Patient Details:

The "reg.html" page with CCS applied used to load the datasets of "Patients Reviews on particular drug"

3.2.2.3. The Patient Report:

The "result.html" page with CCS applied used to display the "Recommendation Report" using given dateset and using python file "app.py" has machine learning implementation.

3.2.2.4. Python File Machine Learning Implemented:

The "app.py" is python file, where it uses "flask" framework to bring values from "reg.html"i.e., "Patient review". The Review will be evaluated by "Machine Learning– RNN bi directional lstm" and predicts the Recommended drug for given input.

3.3. SYSTEM ANALYSIS METHODS

3.3.1. USE CASE DAIGRAM

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

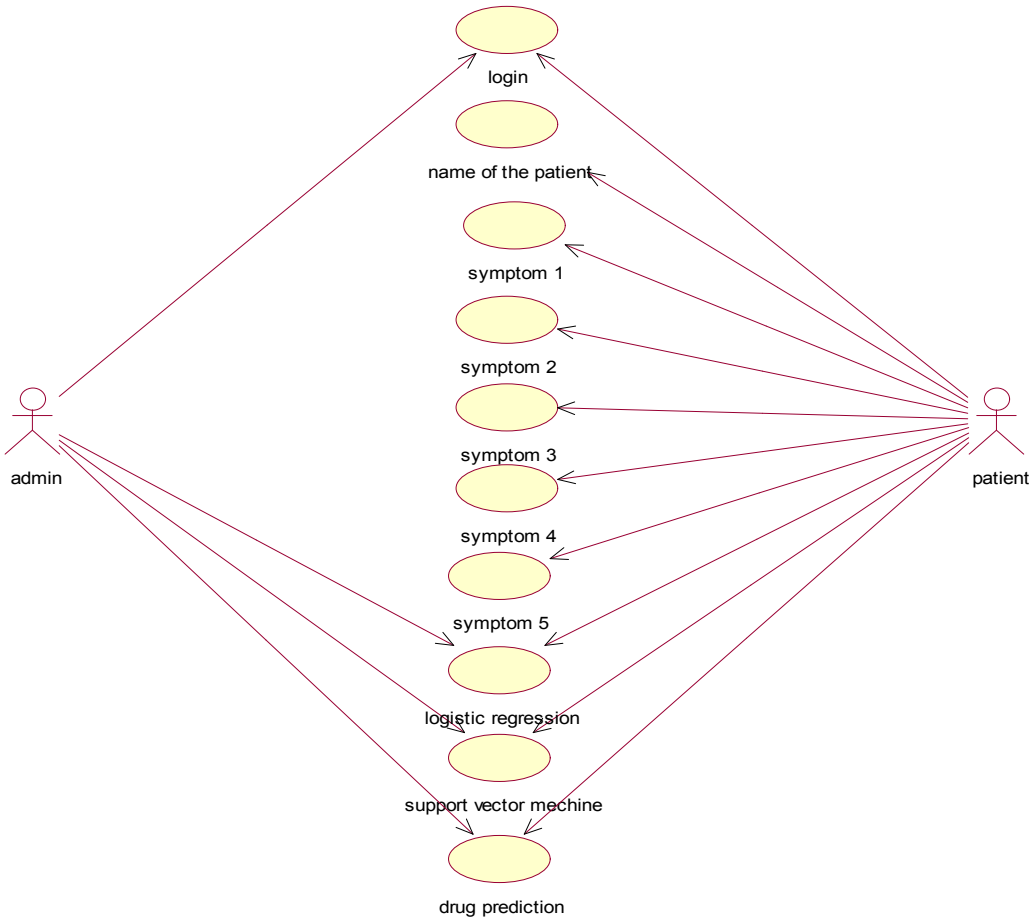


Fig: 3.3.1. UseCaseDiagram

3.3.2. DATAFLOW DIAGRAM

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams.

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel,unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data, flows as a unified model.

Fig 3.3.2 Dataflow Diagram

3.4. SYSTEM DESIGN

System analysis is a problem-solving method that involves looking at the wider system breaking apart the parts, and figuring out how it works in order to achieve a particular goal the main functionality of software is to be responsible for monitoring the changes in attached hardware and to initiate controlling statements. The world has seen tremendous growth in technology during the last decade. As a result of this, a multitude of new applications of the Information & Communication Technologies have emerged and changed the way we live, work, play, interact and even think. Systems are created to solve problems. A collection of components that work together to realize some objectives forms a system. Basically, there are three major components in every system, namely input, processing and output.

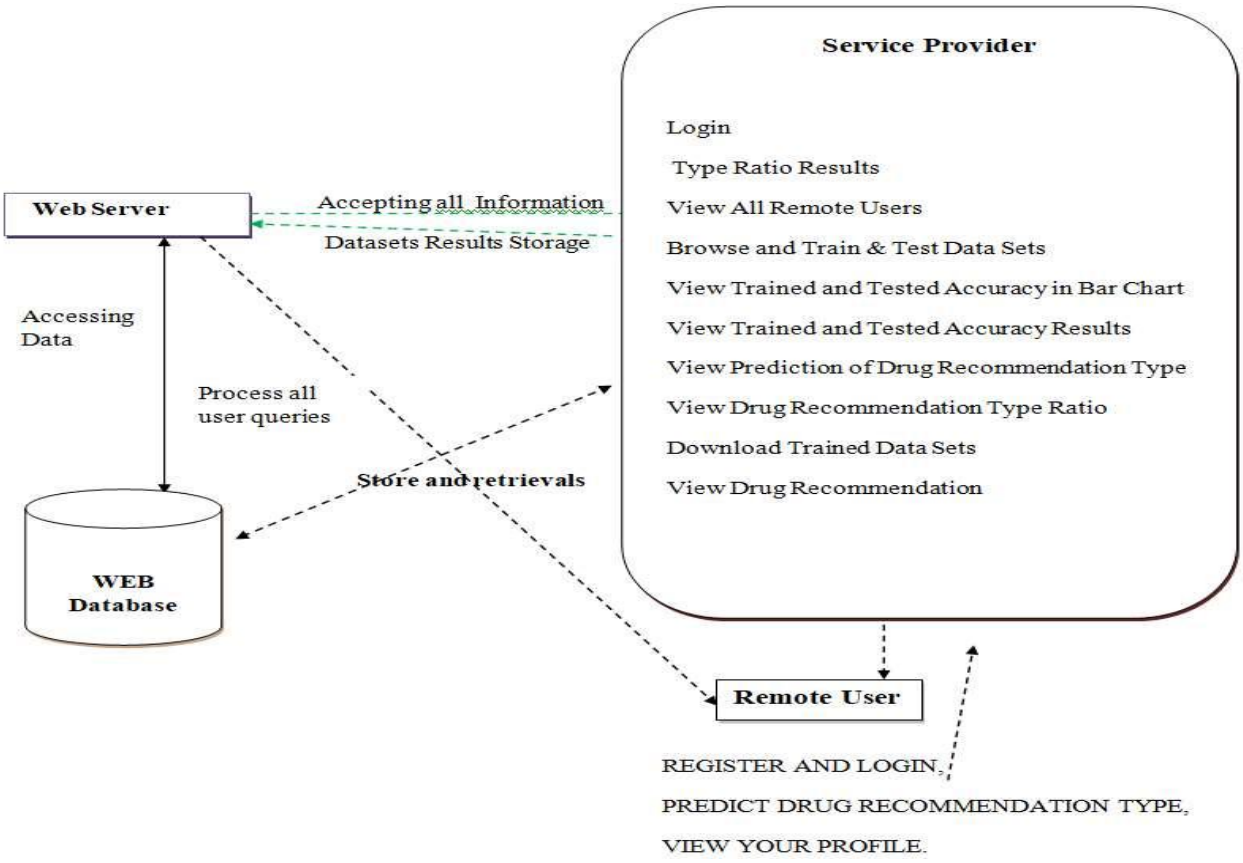


Fig 3.4.1. System Architecture

3.4.1. CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.There are mainly six attributes used in this drug recommendation system namely id,drug name, user's review, condition,rating,date,useful count

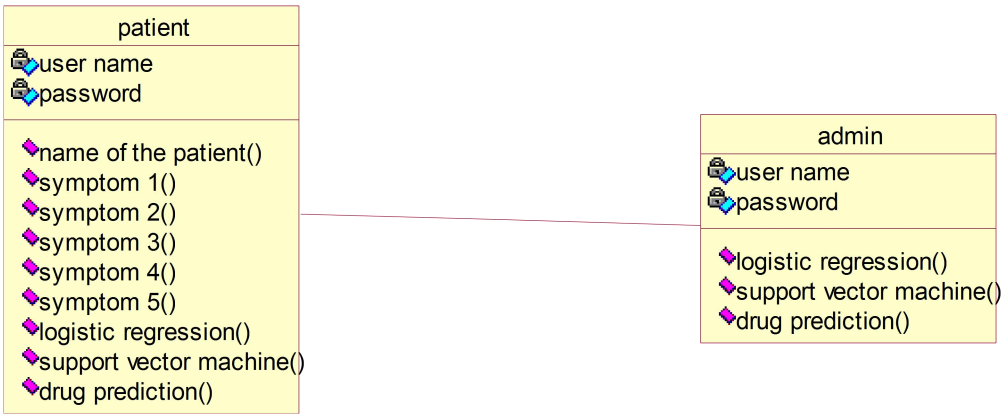


Fig 3.4.2. Class Diagram

3.4.2. SEQUENCE DIAGRAM

The Sequence diagram of the project drug recommendation system based on sentiment analysis of drug reviews using machine learning consist of all the various aspects a normal sequence diagram requires. This sequence diagram shows how from starting the model flows from one step to another, all the information's and all other general information along with the symptoms that goes into the system,compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the sequence of all the entities is linked to each other where the user gets started with the system.

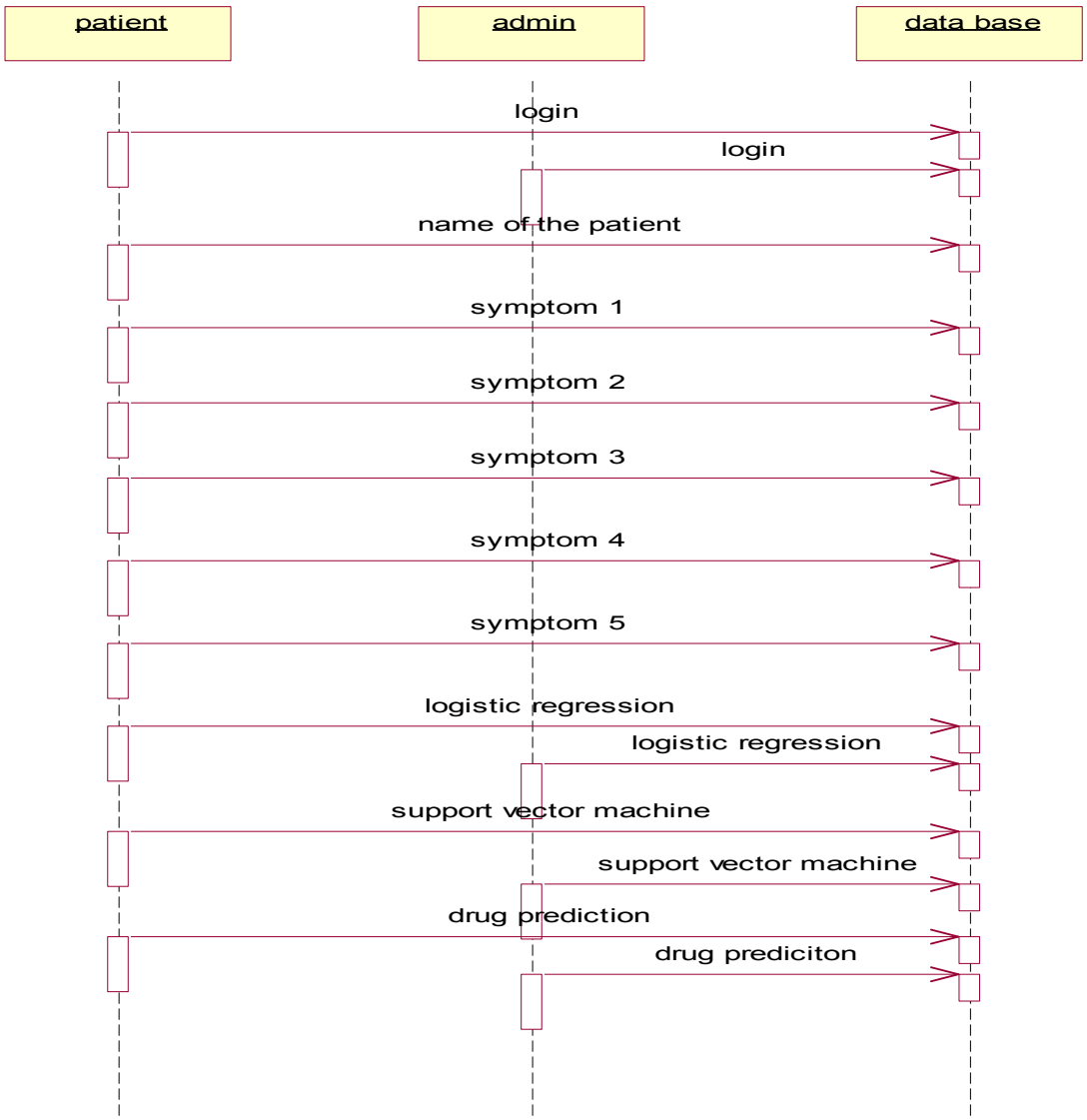


Fig 3.4.3. Sequence Diagram

CHAPTER – 4

IMPLEMENTATION

4. IMPLEMENTATION

4.1. TOOLS USED

4.1.1. ANACONDA

Anaconda Individual Edition contains conda and Anaconda Navigator, as well as Python and hundreds of scientific packages. When you installed Anaconda, you installed all these too.

Conda works on your command line interface such as Anaconda Prompt on Windows and terminal on macOS and Linux.

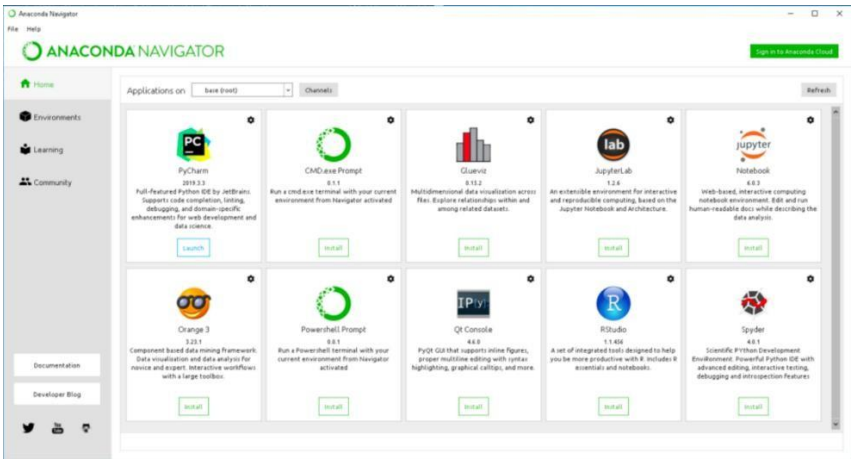
Navigator is a desktop graphical user interface that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands.

You can try both conda and Navigator to see which is right for you to manage your packages and environments. You can even switch between them, and the work you do with one can beviewed in the other.

ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

To get Navigator, get the Navigator Cheat Sheet and install Anaconda. The Getting started with Navigator section shows how to start Navigator from the shortcuts or from a terminal window.



Use of Anaconda Navigator:

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

Applications of Anaconda Navigator:

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- Spyder
- PyCharm
- VSCode
- Glueviz
- Orange 3 App
- RStudio
- Anaconda Prompt (Windows only)
- Anaconda Power Shell (Windows only)

Advanced conda users can also build their own Navigator applications.

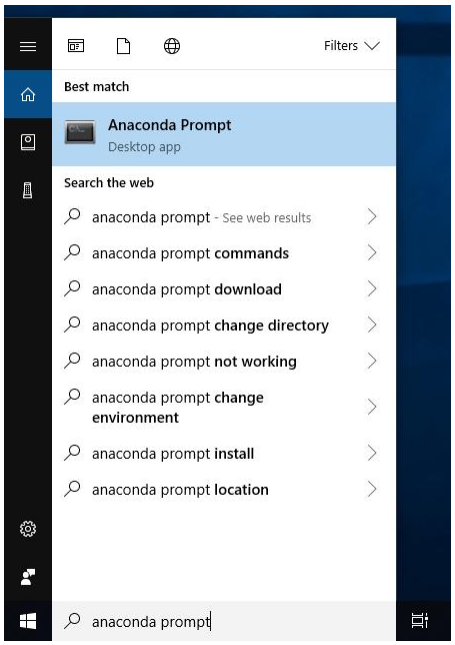
ANACONDA PROMPT

Anaconda Prompt is a command line shell (a program where you type in commands instead of using a mouse). The black screen and text that makes up the **Anaconda Prompt** doesn't look like much, but it is really helpful for problem solvers using Python.

If you prefer using a command line interface (CLI), you can use conda to verify the installation using Anaconda Prompt on Windows or terminal on Linux and macOS.

To open Anaconda Prompt:

- Windows: Click Start, search, or select Anaconda Prompt from the menu.



4.1.2 VISUAL STUDIO CODE

Visual Studio Code supports a wide range of programming languages, including C++, C#, Java, Python, JavaScript, Type Script, and many others. It features built-in support for version control systems like Git and supports various extensions that can be downloaded from the Visual Studio Code Marketplace to enhance its functionality.

Visual Studio Code provides an intuitive user interface, customization keyboard shortcuts, and a powerful integrated terminal for executing commands within the editor. It also features debugging tools that allow developers to debug code directly within the editor and provides rich language support features such as syntax highlighting, code completion, and automatic indentation.

FEATURES OF VISUAL STUDIO CODE

The following is a list of features of visual studio code:

- Intelligent coding assistance: Visual Studio Code provides advanced coding assistance, including syntax highlighting, auto-completion, and code navigation.
- Debugging tools: Visual Studio Code features powerful debugging tools that enable developers to debug their code directly within the editor.
- Extensible: Visual Studio Code is highly extensible, with a rich ecosystem of extensions that can be installed from the Visual Studio Code Marketplace.
- Integrated terminal: Visual Studio Code includes an integrated terminal, allowing developers to execute commands within the editor without having to switch to a separate terminal windows.
- Multi-platform support: Visual Studio Code is available on Windows, macOS, and Linux, making it a truly cross-platform code editor.

4.1.2. HEROKU CLOUD SERVICE

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js , Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages.

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

Heroku works with a wide variety of customers and partners. Learn more about how we support digital and software development agencies, partners, and enterprise companies.

4.2. PSEUDO CODE

CODE FOR RECOMMENDATION :

app.py

```
from flask import Flask, render_template, URL_for, request
import numpy as np
import pickle
import pandas as pd
from keras.models import load_model
from tensor flow.keras.prepossessing.text import Tokenizer
from TensorFlow.keras.prepossessing.sequence import pad_sequences
from keras.models import model_from_json
from numpy import array

app = Flask(__name__)
model = load_model("rnn_model.h5")


with open('tokenizer.pickle', 'rb') as handle:
    tokenizer = pickle.load(handle)


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


@app.route('/predict', methods=['POST'])
def predict():
```

```
 max_length = 200
 if request.method == 'POST':
  review = request.form['review']
  data = [review]
  tokenizer.fit_on_texts(data)
  enc = tokenizer.texts_to_sequences(data)
  enc = pad_sequences(enc, maxlen=max_length, padding='post')
  my_prediction = model.predict(array([enc][0]))[0][0]
      #class1 = model.predict(array([enc][0]))[0][0]


   class1 = (model.predict(enc) > 0.5).astype("int32")


    return render_template('result.html', prediction=class1)

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

app2.py

```
############## Customer Lifetime Value Prediction ###################
from SK learn.feature_extraction.text import Count Vectorizer, TfidfVectorizer
from bs4 import Beautiful Soup
from nltk.corpus import stopwords
import pickle
from sklearn.pipeline import Pipeline
from sklearn.model_selection import Grid Search CV
from sklearn.linear_model import Logistic Regression
from sklearn.metrics import rocauc score, accuracy_score
from sklearn import metrics
from collections import Counter
from imblearn.over_sampling import SMOTE
from sklearn.naive_bayes import BernoulliNB, MultinomialNB
from nltk import sent_tokenize, word_tokenize, postage
from nltk.stem import SnowballStemmer, WordNetLemmatizer
from nltk.stem.porter import PorterStemmer
import nltk
import re
from sklearn.model_selection import train_test_split
import plotly.express as PX
```

```
#loading libraries
import pandas as pd
import numpy as np
import matplotlib.py plot as plt
import seaborn as sns
# %matplotlib inline
import warnings
warnings.filter warnings("ignore")
#loading data
df = pd.read_csv("D:\prooo\drugs Com Test_raw.tsv" error_bad_lines=False, sep='\t'

####" DATA PREPROCESSING "'###

df.isna().sum()

print("Summary statistics of numerical features : \n", df.describe())

print("==========================================================")

print("\n Total number of reviews: ", Len(df))

print("==========================================================")

print("\n Total number of brands: ", Len(list(set(df['drug Name']))))

print("==========================================================")

print("\n Total number of unique products: ", Len(list(set(df['condition']))))

print("==========================================================")

print("\n Percentage of reviews with neutral sentiment : {:.2f}%".format(
   df[df['rating'] == 3]["review"].count()/len(df)*100))

print("==========================================================")

print("\n Percentage of reviews with positive sentiment : {:.2f}%".format(
   df[df['rating'] > 3]["review"].count()/len(df)*100))

print("==========================================================")

print("\nPercentage of reviews with negative sentiment : {:.2f}%".format(
   df[df['rating'] < 3]["review"].count()/len(df)*100))
print("==========================================================")

fig = PX.bar(df['rating'].value_counts().sort_index(),
          x='rating', title='Distribution of Rating')
fig.show()
```

```
drug = df["drugName"].value_counts()
fig = PX.bar(drug[:20], x='drug Name',
        title='Number of Reviews for Top 20 Drugs')
fig.show()

conditions = df["condition"].value_counts()
fig = PX.bar(conditions[:30], x='condition',
        title='Number of Reviews for Top 30 conditions')
fig.show()

review_length = df["review"].dropna().map(lambda x: Len(x))
review_length = review_length.loc[review_length < 1500]
review_length = pd.Data Frame(review_length)
fig = px.histogram(review_length, x="review")
fig.show()

###" DATA PREPARATION"'###

df = df.sample(franc=1, random_state=0)  # un comment to use full set of data

# Drop missing values
df.dropna(in place=True)

# Encode 4s and 5s as 1 (positive sentiment) and 1s and 2s as 0 (negative sentiment)
df['Sentiment'] = np.where(df['rating'] > 6, 1, 0)
df.head()

###'" TRAIN TEST SPLIT"'###

X_train, X_test, y_train, y_test = train_test_split(df['review'], df['Sentiment'],
                                    test_size=0.1, random_state=0)

print('Load %d training examples and %d validation examples. \n' %
     (X_train.shape[0], X_test.shape[0]))
print('Show a review in the training set : \n', X_train.iloc[10])
X_train, y_train

def clean Data(raw_data, remove_stopwords=False, stemming=False,
split_text=False):
   text = Beautiful Soup(raw_data, 'html.parser').get_text()
   letters_only = re.sub("[^a-zA-Z]", " ", text)
   words = letters_only.lower().split()

   if remove_stopwords:
      stops = set(stopwords.words("English"))
      words = [w for w in words if not w in stops]
```

```
if stemming == True:

        steamer = SnowballStemmer('english')
        words = [steamer.stem(w) for w in words]

    if split_text == True:
        return (words)

    return(" ".join(words))

    X_train_cleaned = []
    X_test_cleaned = []

    for d in X_train:
        X_train_cleaned.append(clean Data(d))
    print('Show a cleaned review in the training set : \n',  X_train_cleaned[10])

    for d in X_test:
        X_test_cleaned.append(clean Data(d))


Count Vect = CountVectorizer()
X_train_countVect = countVect.fit_transform(X_train_cleaned)
print("Number of features : %d \n" %
      len(countVect.get_feature_names()))  # 6378
print("Show some feature names : \n", countVect.get_feature_names()[::1000])

smote = SMOTE()
x_train_smote, y_train_smote = smote.fit_sample(X_train_countVect, y_train)
print("Before smote:", Counter(y_train))
print("After smote:", Counter(y_train_smote))

mnb = MultinomialNB()
mnb.fit(X_train_countVect, y_train)

def model Evaluation(predictions):
print("\n Accuracy on validation set: {:.4f}".format(accuracy_score(y_test,
predictions)))
print("\nAUC score : {:.4f}".format(roc_auc_score(y_test, predictions)))
print("\n Classification report : \n",metrics.classification_report(y_test, predictions))
print("\nConfusion Matrix : \n", metrics.confusion_matrix(y_test, predictions))
predictions = mnb.predict(countVect.transform(X_test_cleaned))
Model Evaluation(predictions)
```

```
### TfidfVectorizer with Logistic Regression###
tfidf = TfidfVectorizer(min_df=5)  # minimum document frequency of 5
X_train_tfidf = tfidf.fit_transform(X_train)
print("Number of features : %d \n" % Len(tfidf.get_feature_names()))  # 1722
print("Show some feature names : \n", tfidf.get_feature_names()[::1000])

Lr = Logistic Regression()
Lr.fit(X_train_tfidf, y_train)

feature_names = np.array(tfidf.get_feature_names())
sorted_coed_index = lr.coef_[0].argsort()
print('\n Top 10 features with smallest coefficients :\n{}\n'.format(
   feature_names[sorted_coef_index[:10]]))
print('Top 10 features with largest coefficients : \n{}'.format(
   feature_names[sorted_coef_index[:-11:-1]]))
predictions = Lr.predict(tfidf.transform(X_test_cleaned))
model Evaluation(predictions)
estimators = [("tfidf", TfidfVectorizer()), ("Lr", Logistic Regression())]
model = Pipeline(estimators)
params = {"Lr__C": [0.1, 1, 10],
        "tfidf__min_df": [1, 3],
        "tfidf__max_features": [1000, None],
        "tfidf__ngram_range": [(1, 1), (1, 2)],
        "tfidf__stop_words": [None, "English"]}

grid = Grid Search CV(estimator=model, param_grid=params,
                scoring="accuracy", n_jobs=-1)
grid.fit(X_train_cleaned, y_train)
print("The best parameter set is : \n", grid.best_params_)


# Evaluate on the validation set
predictions = grid.predict(X_test_cleaned)
model Evaluation(predictions)

pickle.dump(mnb, open('Naive_Bayes_model.pkl', 'wb'))
```

Home.html

```
<html>
<head>
<meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/4.4.1/jquery.min.js"></script>
  <script
```

```
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></scr
ipt>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
  <div>
<nav class="navbar navbar-expand-sm bg-dark fixed-top">
  <a class="navbar-brand" href="#"><b>Drug Review Sentiment Analysis</b></a>
  <a class="navbar-brand" href="#"><b><font size="2"></font></b></a>
</nav>
</div>
</head>
<style>
@import
URL('https://fonts.googleapis.com/css2?family=Roboto:wght@100&display=swap');

</style>
<br><br>
<div class="background1">
  <div class="background2">
    <div class="background3">
      <div class="content">
          <center><div class="monospace"><b></b></div></center>
        <center><div class="monospace"><font
size="5"><b></b></font></div></center>
        <br>
          <p style="text-align:left;"><font size="5"><b>Project
Description</b></font></p>

        <div class="text">Sentiment Analysis also known as Opinion Mining refers to
the use of natural language
processing, text analysis to systematically identify, extract, quantify, and study
affective
states and subjective information.</div><br> <div class="text">Sentiment analysis is
widely applied to reviews and
survey responses, online and social media, and healthcare materials for applications
that
range from marketing to customer service to clinical medicine.</div><br>
<div class="text">In this project, we aim to perform Sentiment Analysis of Drug
reviews. Data used
in this project are online product reviews collected from “amazon.com”. We expect to
do
review-level categorization of review data with promising outcomes.</div>
        <br><br>
      </div>
      </div>
    </div>
  </div>
  <div class="social">
```

```
  </div>
</div>
<center><div class="website"><font color="white" size="6"><b>Welcome My
Project</b></font></div></center>
<div class="wrapper">
   <div class="main-div main-div1">

  <h1>Drug Review Sentiment Analysis</h1>
   </div><br>
<br>
<body>
<br>
</div>
        <div class="ml-container">

                <center><form action="{{ URL_for('predict')}}" method="POST">
                <h1 class="text-red blink-hard">Enter Your Review Here</h1>

                <text area name="review" rows="6" cols="50" required="required"
id="ip4"></text area><br/<br><br>
<input type="submit" class="btn bn-primary btn-lg"
value="predict"><br><br><br><br>
        </form></center>
</body>
 </html>
```

Result.html

```
<!DOCTYPE html>
<html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="style/css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/4.4.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></scr
ipt>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<title>Home</title>
        <!-- <link rel="stylesheet" type="text/css" href="../static/css/styles.css"> -->
        <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/styles.css') }}">
```

```
</head>
<body>
<div>
<!--navbar portion-->
  <nav class="navbar navbar-expand-sm bg-dark fixed-top">
  <a class="navbar-brand" href="#"><b>Drug Review Sentiment Analysis</b></a>
  <a class="navbar-brand" href="#"><b><font size="2"></font></b></a>
  <ul class="navbar-nav" >
    <li class="nav-item">
    </li>
    <li class="nav-item">
    </li>
  </ul>
</nav>
<br>
</div>
<body>
<br><br>
<h2 id="ip3"><marquee BEHAVIOR=ALTERNATE direction = "right">Drug
Review Sentiment Analysis</marquee></h2>
</div>
<body>
<div id="drugname">
<!--<h2><center>ML App</center></h2>-->
</div>

        <p style="color:white;font-size:20;text-align: center;"><b>Results for
Comment</b></p>
        <div class="results" "w3-container w3-card " style= "border-
radius:58px;background-color:rgb(223, 230, 226);padding:40px;width: 1350px;"
id="ip4">

        {% if prediction == 1%}
        <h2 style="color:blue">Good Drug.This Drug is Advisable </h2>
        {% elf prediction == 0%}
        <h2 style="color:red;">Bad Drug,This Drug is not Advisable</h2>
        {% endif %}

      </div>
        </div>

</body>
</html>
```

4.3. COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system.Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development. Here component diagram consists of all major components that is used to build a system. So, design, Algorithm,File System and Datasets all are linked to one another.

Datasets are used to compare the results and algorithm is used to process those results and give a correct accuracy and design UI issued to show the result in an appropriate way in the system and file system is used to store the user data. So, like this all components are inter linked to each other.

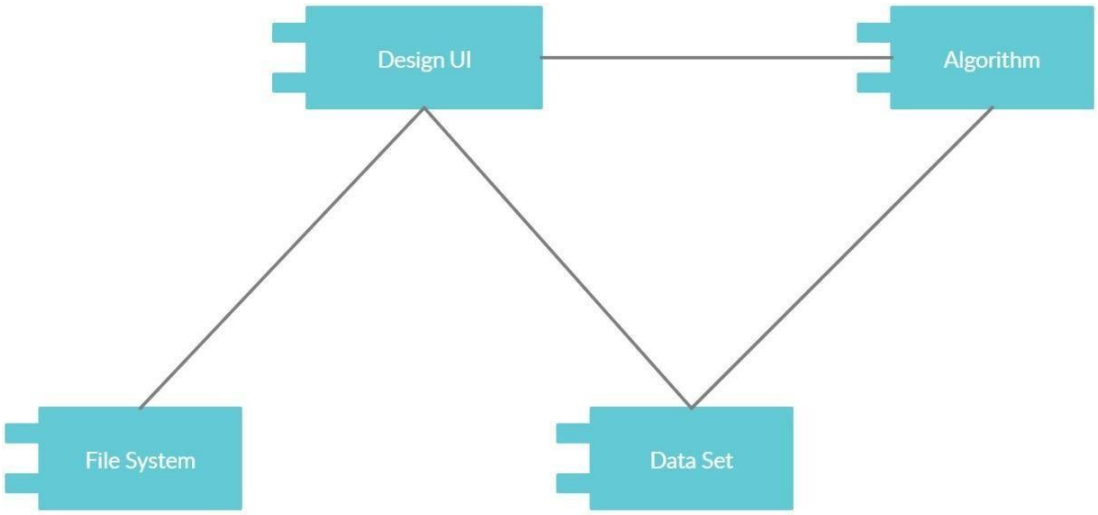


Fig 4.3.1 Component Diagram

4.4. DEPLOYMENT DIAGRAM

A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modelling the physical aspects of an object-oriented system. Here the deployment diagram show the final stage of the project and it also shows how the model looks like after doing all the processes and deploying in the machine. Starting from the system how it processes the user entered information and then comparing that information with the help of datasets, then training and testing those data using the algorithms such as decision tree, naive Bayes, random forest,RNN directional lstm.Then finally processing all those data and information the system gives the desired result in the interface.

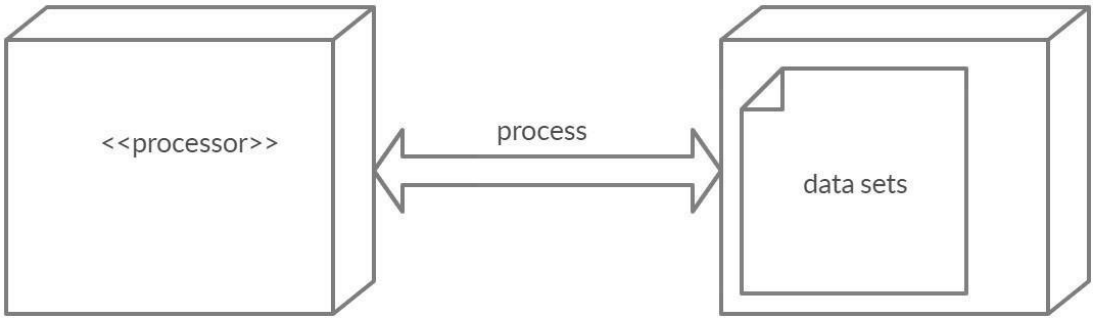


Fig 4.4.1 Deployment Diagram

CHAPTER -5
SCREEN SHOTS

5. SCREEN SHOTS

Fig: 5.1. Login Page

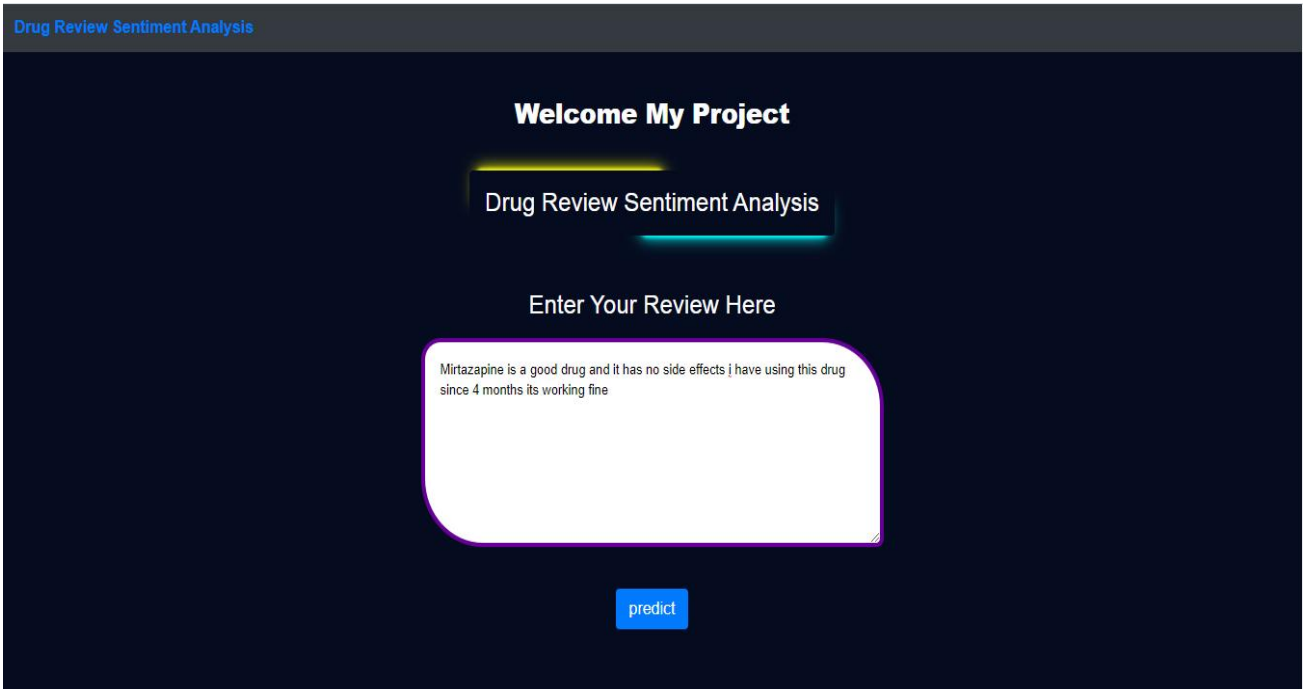


Fig: 5.2.User giving review for prediction

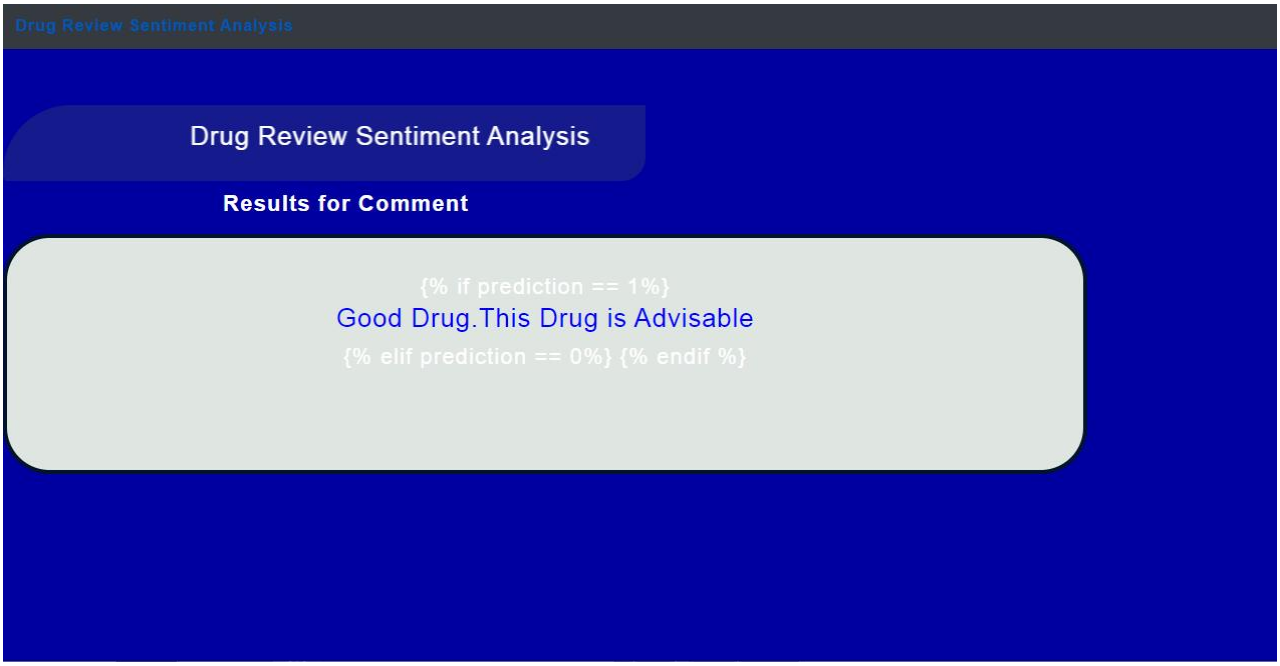


Fig: 5.3 prediction of user's review after sentiment analysis

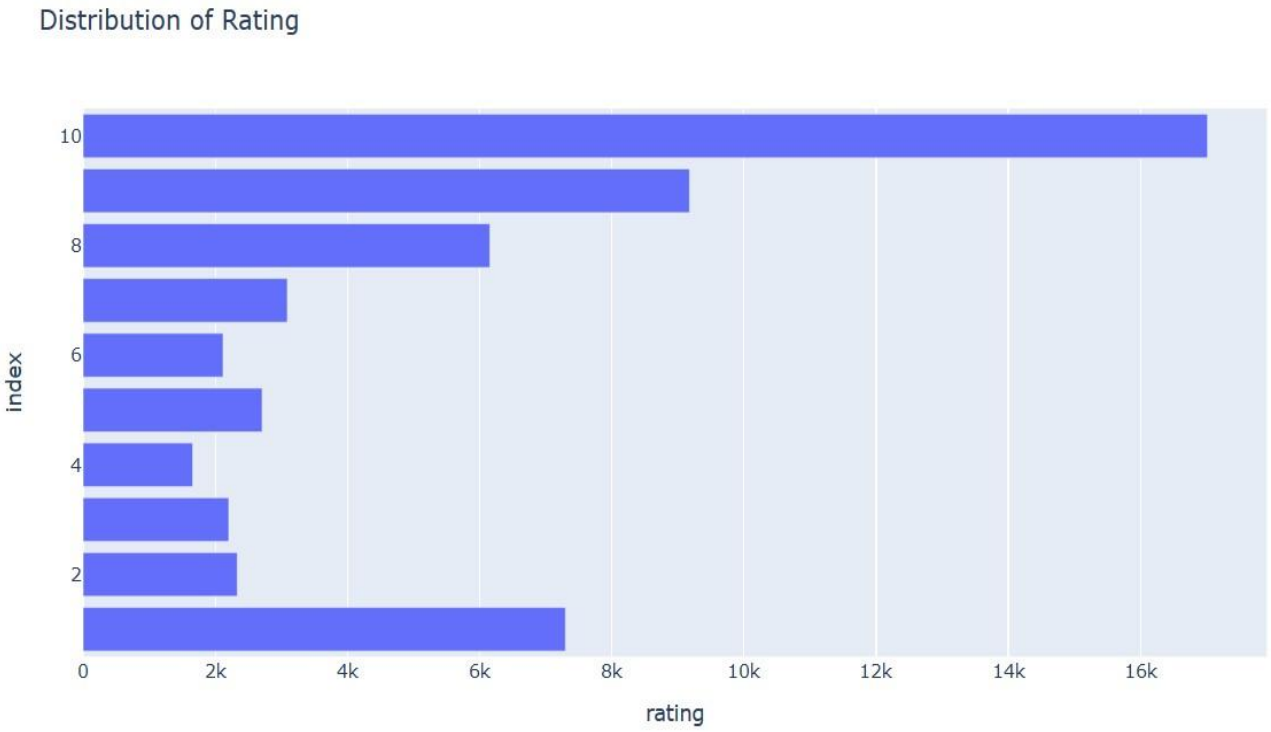


Fig: 5.4 Bar plot showing distribution rating

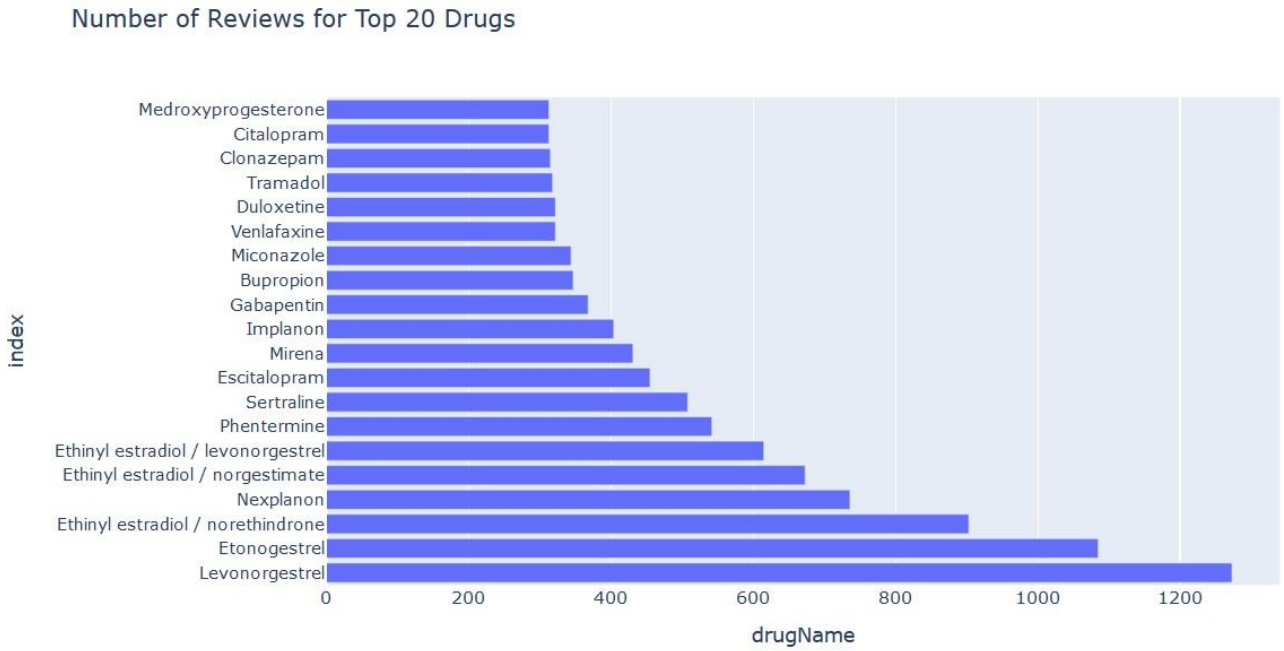


Fig: 5.5.No of Reviews for top 20 Drugs.

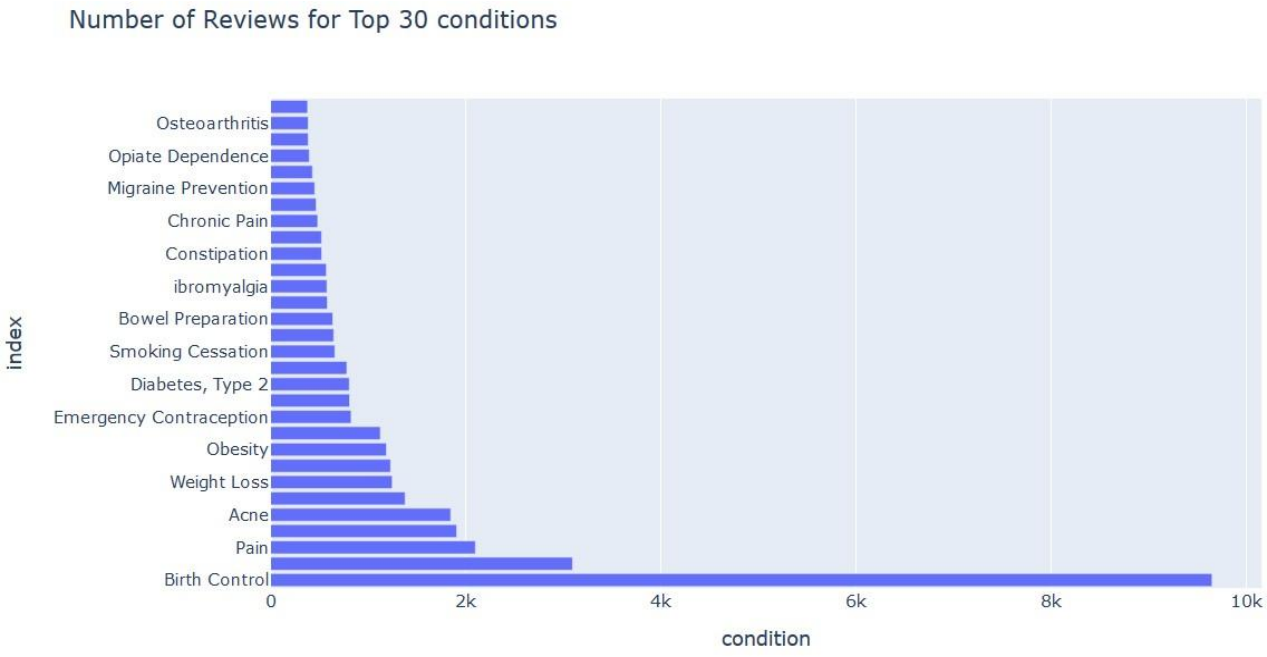


Fig: 5.5. No of Reviews for top 30 Conditions

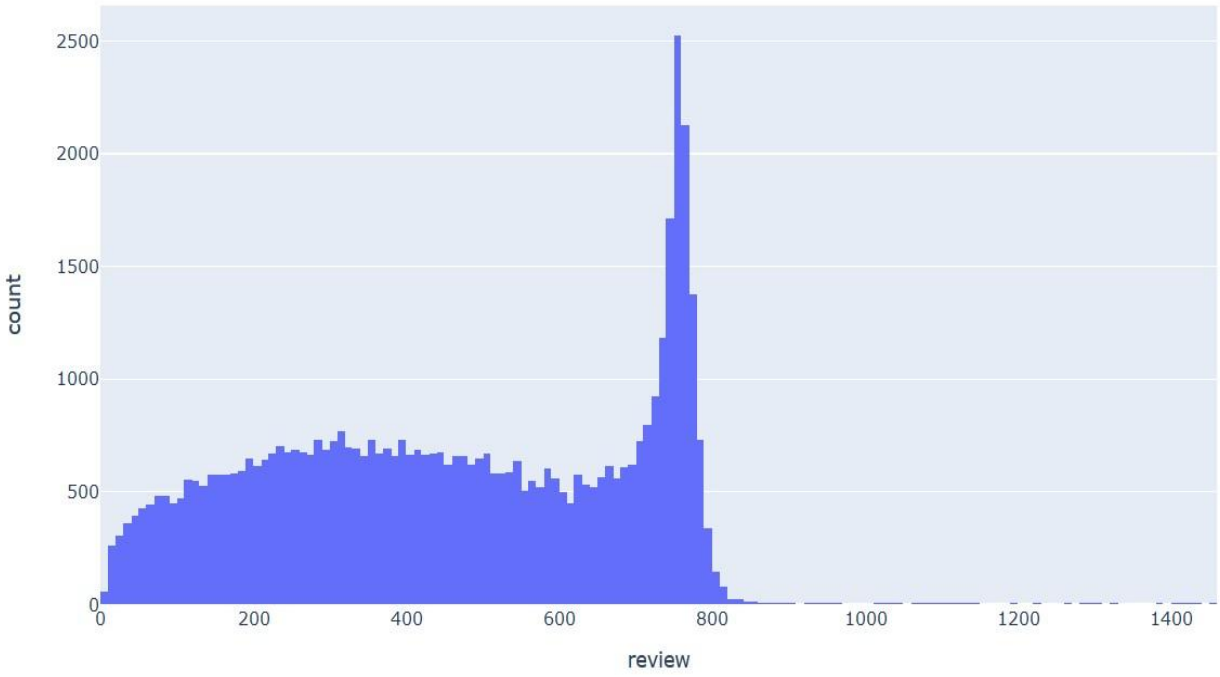


Fig: 5.6 Useful count of reviews

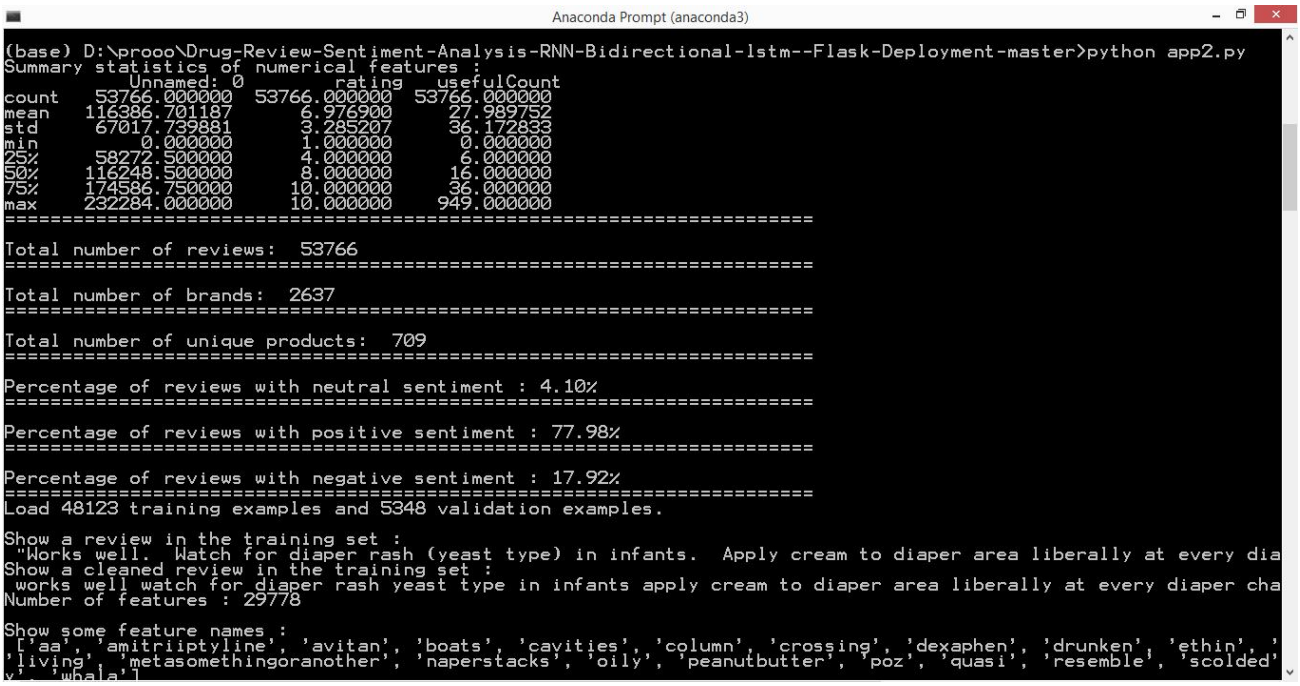


Fig: 5.7 Showing the results after performing all the necessary methodologies

CHAPTER -6

TESTING

6. TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system.

TYPES OF TESTING

6.1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application.It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results

6.2. INTEGRATION TESTING

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

6.3. VALIDATION TESTING

Validation testing is a type of software testing that is performed to ensure that the software product meets the requirements and specifications of the end-users. The main purpose of validation testing is to check whether the software system is fulfilling its intended use and delivering the expected results. It is a process of evaluating the software product to ensure that it meets the customer's expectations and requirements.

In validation testing, the software system is tested as a whole to ensure that all the components are working together as expected, and the system is delivering the desired results. This type of testing is usually performed at the end of the software development life cycle (SDLC) after the completion of the testing phase..

Validation is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

The testing process overview is as follows:

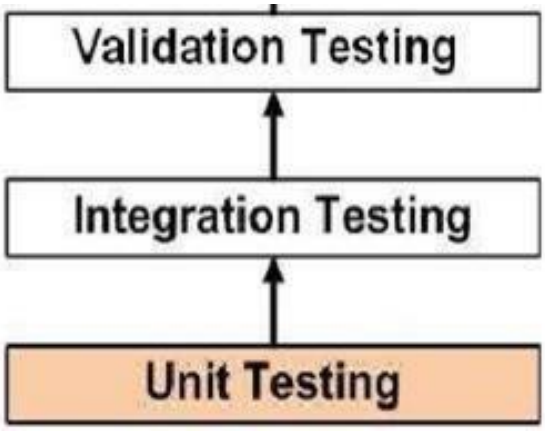


Fig 6.1 The Testing Process

6.4. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within thescope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.As a rule, system testing takes, as its input, all of the "integrated" software components that havesuccessfully passed integration testing and also the software system itself integrated with any applicable hardware system.

System testing is a more limited type of testing; it seeks to detect defectsboth within the "inter-assemblages" and also within the system as a whole. System testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) or System Requirement Specification (SRS).

6.5. TESTING OF INITIALIZATION AND UI COMPONENTS

Serial Number of Test Case	TC 01
Module Under Test	Login Credentials
Description	The user or Doctor enters to login into the Patient's Registration Form.
Input	Username and Password.
Output	If the User's details are correct, they will be signed in. If not, it displays credentials are incorrect.
Remarks	Test Successful.

Table 6.5.1 Test Case for User Registration

Serial Number of Test Case	TC 02
Module Under Test	User Login
Description	When the user tries to log in, details of user are verified in the system.
Input	Username and Password.
Output	If the login details are correct, the user is logged in and user page is displayed. If the login details are incorrect, Displays error message.
Remarks	Test Successful.
Error	Check Credentials

Table 6.5.2 Test Case for User Login

Serial Number of Test Case	TC 03
Module Under Test	Prediction Result
Description	User needs to enter the reviews to get the prediction result.
Input	Drug Name and Symptoms
Output	After applying the sentiment analysis for the user's review,then the predicted result will be obtained.
Remarks	Test Successful.

Table 6.5.3 Test Case for Prediction Result

CHAPTER -7

SUMMARY & CONCLUSION

7. SUMMARY & CONCLUSION

- The project involved several stages, including data collection, data preprocessing, feature extraction, sentiment analysis, and visualization of the results

.

- Through the data collection process, a datasets of drug reviews was gathered from various online sources, including social media platforms and healthcare websites. The data was then per-processed to remove noise, irrelevant information, and duplicate records.

- The feature extraction stage involved identifying the key features and attributes of the drug reviews, including the drug name, dosage, side effects, and efficacy. These features were then used to train and test the sentiment analysis model, which was developed using machine learning algorithms such as Naive Bayes and Support Vector Machines.

- The sentiment analysis stage involved analyzing the reviews to determine the sentiment expressed in them, including positive, negative, or neutral sentiments. The results of the sentiment analysis were then visualized using various techniques, such as word clouds, bar charts, and histograms.

- The project can be useful for healthcare professionals, pharmaceutical companies, and consumers in making informed decisions about drugs based on the sentiments expressed in user reviews.

- The project can also be extended to include more advanced sentiment analysis techniques, such as aspect-based sentiment analysis and opinion mining, to improve the accuracy and reliability of the results.

CHAPTER -8

FUTURE ENHANCEMENT

8. FUTURE ENHANCEMENT

- *Expand the dataset:* This could be achieved by scraping more data from online review websites or by partnering with healthcare providers to collect reviews from their patients.

- *Add additional features:* The project could be enhanced by adding additional features to the website, such as user profiles, personalized recommendations, or more detailed drug information. This could help to improve the user experience and make the website more valuable to users.

- *Incorporate user feedback:* The project could be enhanced by incorporating user feedback and ratings into the sentiment analysis model. This could help to improve the accuracy of the model and provide more valuable insights to users.

- *Partner with healthcare providers*: The project could be enhanced by partnering with healthcare providers to provide more comprehensive drug information and analysis. the project as a trusted source of information for patients and healthcare professionals.

- *Improve the sentiment analysis model*: The sentiment analysis model could be improved by using more advanced machine learning algorithms or by incorporating natural language processing (NLP) techniques. This could help to better identify and classify the sentiment of drug reviews, especially those that contain more complex language or nuanced sentiments.

CHAPTER-9

BIBLIOGRAPHY

9. BIBLIOGRAPHY

[1] Telemedicine, https://www.mohfw.gov.in/pdf/Telemedicine.pdf

[2] Wittich CM, Burkle CM, Lanier WL. Medication errors: an overview for clinicians. Mayo Clin Proc. 2014 Aug;89(8):1116-25.

[3] CHEN, M. R., & WANG, H. F. (2013). The reason and prevention of hospital medication errors. Practical Journal of Clinical Medicine, 4.

[4] Drug Review Datasets, https://archive.ics.uci.edu/ml/datasets/Drug%2BReview%2BDataset%2B%2528Drugs.com%2529#

[5] Fox, Susannah, and Mauve Duggan. "Health online 2013. 2013." URL: http://pewinternet.org/Reports/2013/Health-online.aspx

[6] Bartlett JG, Dowell SF, Mandell LA, File TM Jr, Musher DM, Fine MJ. Practice guidelines for the management of community-acquired pneumonia in adults. Infectious Diseases Society of America. Clin Infect Dis. 2000 Aug;31(2):347-82.doi: 10.1086/313954. Epub 2000 Sep 7. PMID: 10987697; PMCID: PMC7109923.

[7] Fox, Susannah & Duggan, Maeve. (2012). Health Online 2013. Pew Research Internet Project Report.

[8] T. N. Tekade and M. Emmanuel, "Probabilistic aspect mining approach for interpretation and evaluation of drug reviews," 2016 International Conference on Signal Processing, Communication, Power and Embed- ded System (SCOPES), Paraphernalia, 2016, pp. 1471-1476, doi: 10.1109/SCOPES.2016.7955684.

[9] Doulaverakis, C., Nikolaidis, G., Kleontas, A. et al. Galen Owl: Ontology-based drug recommendations discovery. J Biomed Semant 3, 14 (2012). https://doi.org/10.1186/2041-1480-3-14

[10] Leilei Sun, Chuanren Liu, Chonghui Guo, Hui Xiong, and Yanming Xie. 2016. Data-driven Automatic Treatment Regimen Development and Recommendation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 1865–1874. DOI:https://doi.org/10.1145/2939672.2939866

[11] V. Goel, A. K. Gupta and N. Kumar, "Sentiment Analysis of Mul- tilingual Twitter Data using Natural Language Processing," 2018 8th International Conference on Communication Systems and Net- work Technologies (CSNT), Bhopal, India, 2018, pp. 208-212, doi: 10.1109/CSNT.2018.8820254.

[12] Shimada K, Takada H, Mitsuyama S, et al. Drug-recommendation system for patients with infectious diseases. AMIA Annu Symp Proc. 2005;2005:1112.

[13] Y. Bao and X. Jiang, "An intelligent medicine recommended system framework," 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA),Hefei,2016,pp.1383-1388,doi: 10.1109/ICIEA.2016.7603801.

[14] Zhang, Yin & Zhang, Dafang & Hassan, Mohammad & Alamri, Atif & Peng, Limei. (2014). CADRE: Cloud-Assisted Drug Recommendation Service for Online Pharmacies. Mobile Networks and Applications. 20. 348-355. 10.1007/s11036-014-0537-4.

[15] J. Li, H. Xu, X. He, J. Deng and X. Sun, "Tweet modeling with LSTM recurrent neural networks for hashtag recommendation," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 1570-1577, doi: 10.1109/IJCNN.2016.7727385.