

DISCOVERING DRUGS TO TREAT VARIOUS DISEASES FOR HUMAN BEINGS

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

By

K.Vamsidhar Reddy(38110235)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119
MARCH 2022**



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of K.Vamsidhar Reddy(38110235) who carried out the project entitled "**DISCOVERING DRUGS TO TREAT VARIOUS DISEASES FOR HUMAN BEINGS**"

under my supervision from November 2021 to March 2022.

Internal Guide

DR. K. MOHANA PRASAD M.Tech, Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I K.Vamsidhar reddy hereby declare that the Project Report entitled “**DISCOVERING DRUGS TO TREAT VARIOUS DISEASES FOR HUMAN BEINGS**” done by us under the guidance of DR.K.MOHANA PRASAD is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in 2018-2022.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D., Dean**, School of Computing **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D.** , Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Dr.K.Mohana Prasad** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

CONTENT		
Chapter	List of content	Page no.
i	Abstract	07
1	Introduction	08
2	Literature Survey	13
3	System Requirements	16
	3.1 Hardware Requirements	16
	3.2 Software Requirements	17
	3.3 Language Specification	17
	3.4 History of python	
	3.5 Application of python	18
	3.6 Features of python	19
	3.7 Feasibility of study	19
	3.7.1 Economical feasibility	19
	3.7.2 Technical feasibility	19
	3.7.3 Social feasibility	20
4	System Analysis	21
	4.1 Purpose	21
	4.2 Scope	21
	4.3 Exsisting System	21
	4.4 Proposed system	22
5	System Design	23
	5.1 Input design	23
	5.2 Output design	23
6	Modules Description	33
7	System implementation	38
	7.1 System architecture	23
8	System testing	39
	8.1 Test plan	39
	8.2 Verification	39
	8.3 Validation	39
	8.4 Basics of software testing	39

8.5	Black box testing	40
8.6	White box testing	40
8.7	Types of testing	40
8.7.1	Unit testing	41
8.7.2	Integration testing	41
8.7.3	Functional testing	41
8.7.4	System testing	41
8.7.5	Stress testing	43
8.7.6	Performance testing	43
8.7.7	Usability testing	43
8.7.8	Acceptance testing	43
8.7.9	Regression testing	43
9	Conclusion	46
iv	References	

DRUG DISCOVERY FOR PROJECT APPLICATION USING MACHINE LEARNING

ABSTRACT

The advancements of information technology and related processing techniques have created a fertile base for progress in many scientific fields and industries. In the fields of drug discovery and development, machine learning techniques have been used for the development of novel drug candidates. The methods for designing drug targets and novel drug discovery now routinely combine machine learning and deep learning algorithms to enhance the efficiency, efficacy, and quality of developed outputs. The generation and incorporation of big data, through technologies such as high-throughput screening and high through-put computational analysis of databases used for both lead and target discovery, has increased the reliability of the machine learning and deep learning incorporated techniques. The use of these virtual screening and encompassing online information has also been highlighted in developing lead synthesis pathways. In this review, machine learning algorithms utilized in drug discovery and associated techniques will be discussed. The applications that produce promising results and methods will be reviewed.

Chapter 1

INTRODUCTION

Machine learning (ML), an essential component in AI, has been integrated into many fields, such as data generation and analytics. The basis of algorithm-based techniques, such as ML, requires a heavy mathematical and computational theory. ML models have been used in many promising technologies, such as deep learning (DL) assisted self-driving cars, advanced speech recognition, and support vector machine-based smarter search engines. The advent of these computer-assisted computational techniques, first explored in the 1950s, has already been used in drug discovery, bioinformatics, cheminformatics, etc. Drug discovery has been based on a traditional approach that focuses on holistic treatment. In the last century, the world's medical communities started to use an allopathic approach to treatment and recovery. This change led to the success of fighting diseases, but high drug costs ensued, becoming a healthcare burden. While quite diverse and specific to candidates, the cost of drug discovery and development has consistently and dramatically increased. The generalized components of early drug discovery include target identification and characterization, lead discovery, and lead optimization. Many computer-based approaches have been used for the discovery and optimization of lead compounds, including molecular docking, pharmacophore modeling, decision forests, and comparative molecular field analysis. ML and DL have become attractive and DL have become attractive approaches to drug discovery. The applications of ML and DL algorithms in drug discovery are not limited to a specific step, but for the whole process. In this article, we review the ML and DL algorithms that have been widely used in drug discovery.

1.1 OBJECTIVE

- Drug discovery is the process by which drugs is aims at identifying a compound therapeutically useful in curing & treating disease.
- The process of drug discovery involves the identification of candidates, synthesis, characterization, screening & assays for therapeutic efficacy.
- A comparison of several machine learning techniques has been performed in order to obtain a satisfactory classifier for detecting drug target articles using semantic information .
- Our objective is to Identify the drug discovery for patients Using SVM and KNN algorithms.

1.2 Proposed Algorithms:

SVM

TfidfVectorizer transformer:

TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.

Term Frequency (TF)

It is a measure of the frequency of a word (w) in a document (d). TF is defined as the ratio of a word's occurrence in a document to the total number of words in a document. The denominator term in the formula is to normalize since all the corpus documents are of different lengths.

$$TF(w, d) = \frac{\text{occurences of } w \text{ in document } d}{\text{total number of words in document } d}$$

Image by Author

EXAMPLE

Documents	Text	Total number of words in a document
A	Jupiter is the largest planet	5
B	Mars is the fourth planet from the sun	8

Image by Author

The initial step is to make a vocabulary of unique words and calculate TF for each document. TF will be more for words that frequently appear in a document and less for rare words in a document.

Words	TF (for A)	TF (for B)
Jupiter	1/5	0
Is	1/5	1/8
The	1/5	2/8
largest	1/5	0
Planet	1/5	1/8
Mars	0	1/8
Fourth	0	1/8
From	0	1/8
Sun	0	1/8

Image by Author

Inverse Document Frequency (IDF)

It is the measure of the importance of a word. Term frequency (TF) does not consider the importance of words. Some words such as 'of', 'and', etc. can be most frequently present but are of little significance. IDF provides weightage to each word based on its frequency in the corpus D.

IDF of a word (w) is defined as

$$IDF(w, D) = \ln\left(\frac{\text{Total number of documents (N) in corpus D}}{\text{number of documents containing w}}\right)$$

Image by Author

In our example, since we have two documents in the corpus, N=2.

Words	TF (for A)	TF (for B)	IDF
Jupiter	1/5	0	$\ln(2/1) = 0.69$
Is	1/5	1/8	$\ln(2/2) = 0$
The	1/5	2/8	$\ln(2/2) = 0$
largest	1/5	0	$\ln(2/1) = 0.69$
Planet	1/5	1/8	$\ln(2/2) = 0$
Mars	0	1/8	$\ln(2/1) = 0.69$
Fourth	0	1/8	$\ln(2/1) = 0.69$
From	0	1/8	$\ln(2/1) = 0.69$
Sun	0	1/8	$\ln(2/1) = 0.69$

Image by Author

Term Frequency — Inverse Document Frequency (TFIDF)

It is the product of TF and IDF.

TFIDF gives more weightage to the word that is rare in, the corpus (all the documents).

TFIDF provides more importance to the word that is more frequent in the document.

$$TFIDF(w, d, D) = TF(w, d) * IDF(w, D)$$

Image by Author

Words	TF (for A)	TF (for B)	IDF	TFIDF (A)	TFIDF (B)
Jupiter	1/5	0	$\ln(2/1) = 0.69$	0.138	0
Is	1/5	1/8	$\ln(2/2) = 0$	0	0
The	1/5	2/8	$\ln(2/2) = 0$	0	0
largest	1/5	0	$\ln(2/1) = 0.69$	0.138	0
Planet	1/5	1/8	$\ln(2/2) = 0$	0.138	0
Mars	0	1/8	$\ln(2/1) = 0.69$	0	0.086
Fourth	0	1/8	$\ln(2/1) = 0.69$	0	0.086
From	0	1/8	$\ln(2/1) = 0.69$	0	0.086
Sun	0	1/8	$\ln(2/1) = 0.69$	0	0.086

After applying TFIDF, text in A and B documents can be represented as a TFIDF vector of dimension equal to the vocabulary words

Chapter 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of

external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

Applications of machine learning in drug discovery and development

Drug discovery and development pipelines are long, complex and depend on numerous factors. Machine learning (ML) approaches provide a set of tools that can improve discovery and decision making for well-specified questions with abundant, high-quality data. Opportunities to apply ML occur in all stages of drug discovery. Examples include target validation, identification of prognostic biomarkers and analysis of digital pathology data in clinical trials. Applications have ranged in context and methodology, with some approaches yielding accurate predictions and insights. The challenges of applying ML lie primarily with the lack of interpretability and repeatability of ML-generated results, which may limit their application. In all areas, systematic and comprehensive high-dimensional data still need to be generated. With ongoing efforts to tackle these issues, as well as

increasing awareness of the factors needed to validate ML approaches, the application of ML can promote data-driven decision making and has the potential to speed up the process and reduce failure rates in drug discovery and development.

A review on machine learning approaches and trends in drug discovery

Drug discovery aims at finding new compounds with specific chemical properties for the treatment of diseases. In the last years, the approach used in this search presents an important component in computer science with the skyrocketing of machine learning techniques due to its democratization. With the objectives set by the Precision Medicine initiative and the new challenges generated, it is necessary to establish robust, standard and reproducible computational methodologies to achieve the objectives set. Currently, predictive models based on Machine Learning have gained great importance in the step prior to preclinical studies. This stage manages to drastically reduce costs and research times in the discovery of new drugs. This review article focuses on how these new methodologies are being used in recent years of research. Analyzing the state of the art in this field will give us an idea of where cheminformatics will be developed in the short term, the limitations it presents and the positive results it has achieved. This review will focus mainly on the methods used to model the molecular data, as well as the biological problems addressed and the Machine Learning algorithms used for drug discovery in recent years.

Artificial intelligence to deep learning: machine intelligence approach for drug discovery

Drug designing and development is an important area of research for pharmaceutical companies and chemical scientists. However, low efficacy, off-target delivery, time consumption, and high cost impose a hurdle and challenges that impact drug design and discovery. Further, complex and big data from genomics, proteomics, microarray data, and clinical trials also impose an obstacle in the drug discovery pipeline. Artificial intelligence and machine learning technology play a crucial role in drug discovery and development. In other words, artificial neural networks and deep learning algorithms have modernized the area. Machine learning and deep learning algorithms have been implemented in several drug discovery processes such as peptide synthesis, structure-based virtual screening, ligand-based virtual screening, toxicity prediction, drug monitoring and release, pharmacophore modeling, quantitative structure–activity relationship, drug repositioning, polypharmacology, and physiochemical activity. Evidence from the past strengthens the implementation of artificial intelligence and deep learning in this field. Moreover, novel data mining, curation, and management techniques provided critical support to recently developed modeling algorithms. In summary, artificial intelligence and deep learning advancements provide an excellent opportunity for rational drug design and discovery process, which will eventually impact mankind.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

System : Pentium i3 Processor

Hard Disk : 500 GB.

Monitor : 15'' LED

Input Devices : Keyboard, Mouse

Ram : 2 GB

3.2 SOFTWARE REQUIREMENTS

Operating system : Windows 10

Coding Language : Python

3.3 LANGUAGE SPECIFICATION

Python is a general-purpose interpreted, interactive, object-oriented, and high-level 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). This **tutorial** gives enough understanding on **Python programming** language.

3.4. HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

3.5. APPLICATION OF PYTHON

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

3.6 FEATURES OF PYTHON

- 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.

3.7 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

3.7.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into

the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.7.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.7.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

CHAPTER 4

SYSTEM ANALYSIS

4.1 PURPOSE

The purpose of this document is identification of drug discovery for patients using machine learning algorithms. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project - such as interface, functional requirements and performance requirements.

4.2 SCOPE

The scope of this SRSdocument persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRSto the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

4.3 EXISTING SYSTEM

In existing System, used generative RNN models to develop possible molecular structures that could have activity against Staphylococcus aureus (S. aureus) and Plasmodium falciparum (P. falciparum). Their models were given small sets of molecular structures that had known activity against these target organisms; from these inputs, the model generated 14% of the 6051 potential molecule candidates

for *S. aureus* that has been developed by medicinal chemists. The model also generated 28% of the existing compounds developed for *P. falciparum*.

DISADVANTAGES OF EXISITNG SYSTEM

- Required the biggest associated area to find the medication name.
- Poor Edge identification.
- Not conceivable to group a book.

4.4 PROPOSED SYSTEM

- In our daily lives, many medicines used in hospitals and primary care centers are difficult to identify. And those used may be burdensome to detect on a day-to-day basis unless they are obvious.
- Medicines are one of the most important health care technologies for improving health and quality of life for the generations.
- Initially, a particular model must be characterized along with parameters. So, machines can be gain proficiency in the model with accessible parameters through the utilization of trained data.
- They utilized SVM to create models that could predict drugs and their efficacy.
- Furthermore, the model can discover the drug for patients from the given data.

ADVANTAGES OF PROPOSED SYSTEM

- Accurate detection
- Less Time complexity

CHAPTER 5

INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output

element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

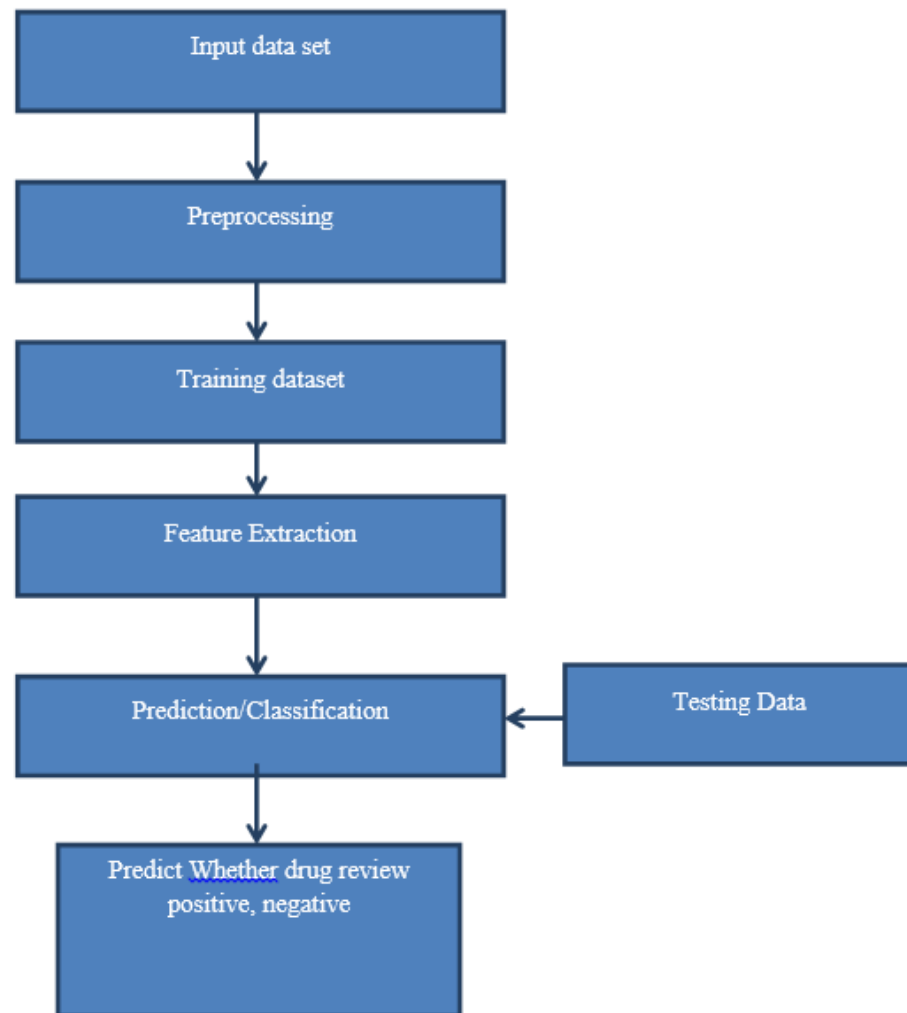
- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

5.3 DATA FLOW DIAGRAM:

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

depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



UML DIAGRAMS

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

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

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

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

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

GOALS:

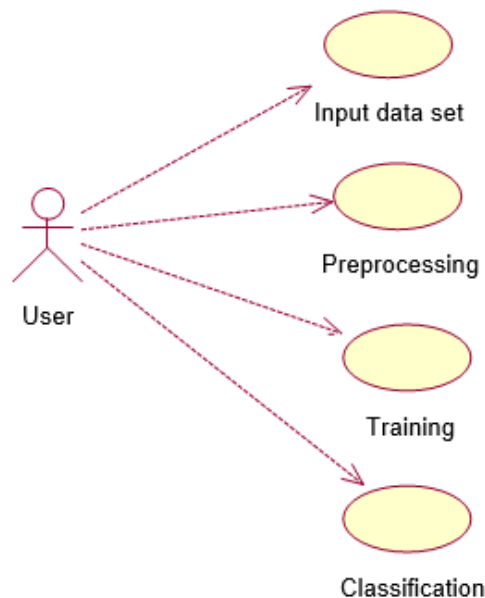
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

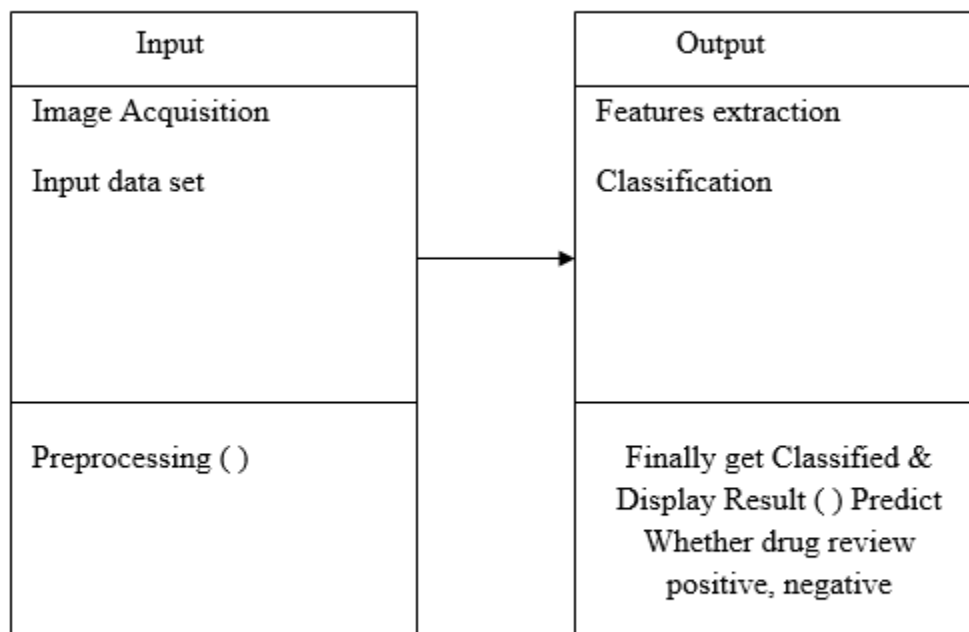
USE CASE DIAGRAM:

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



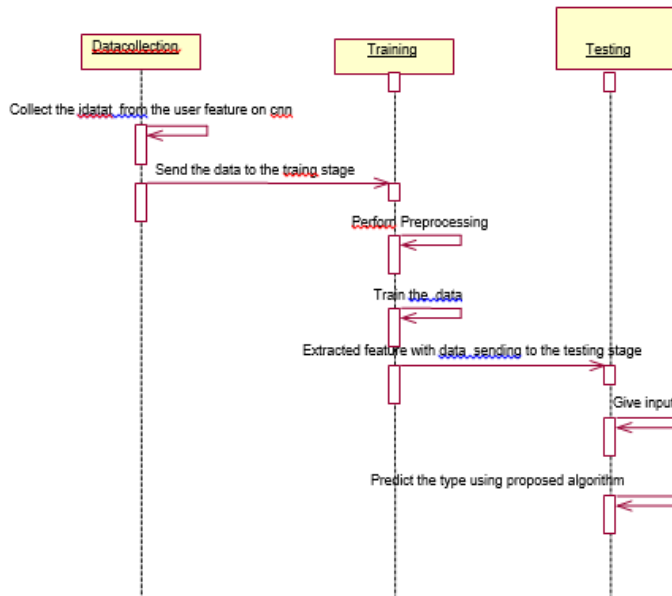
CLASS DIAGRAM:

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.



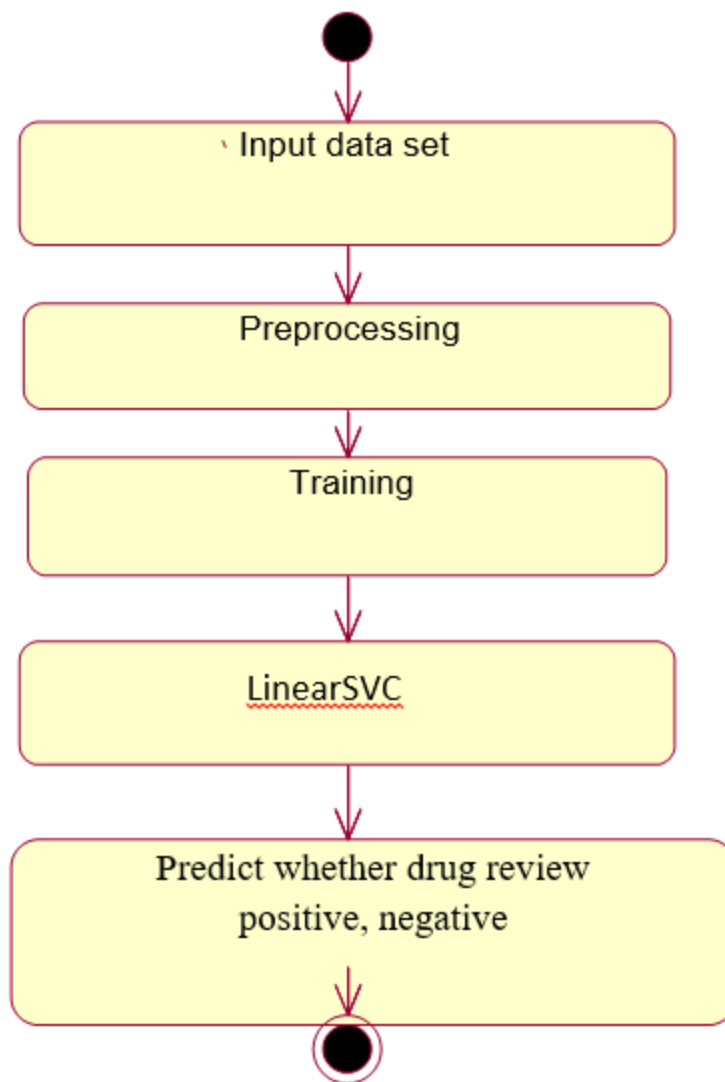
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



CHAPTER 6

MODULES

6.1 MODULES

- Data Collection
- Dataset
- Data Preparation

- Model Selection
- Analyze and Prediction
- Accuracy on test set
- Saving the Trained Model
- Database connecting using My sql

MODULE DESCRIPTION

6.1.1 Data Collection:

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.

There are several techniques to collect the data, like web scraping, manual interventions and etc.

Drug Recommendation System based on Sentiment Analysis of Drug Reviews using Machine Learning

Data set Link: <https://www.kaggle.com/jessicali9530/kuc-hackathon-winter-2018>

6.1.2 Dataset:

The train dataset consists of 161297 and test dataset consists of 53766 There are 3 columns in the dataset, which are described below

Index: unique id

DrugName: Name of drug used

Condition: Condition of a patient

Review : Review of a patient

Rating : 1 to 10

Date : The day, month, or year

usefulCount: review count

6.1.3 Data Preparation:

We will transform the data. by getting rid of missing data and removing some columns. First we will create a list of column names that we want to keep or retain. Next we drop or remove all columns except for the columns that we want to retain. Finally we drop or remove the rows that have missing values from the data set.

6.1.4 Model Selection:

We used Linear SVC. The Linear Support Vector Classifier (SVC) method applies a linear kernel function to perform classification and it performs well with a large number of samples. If we compare it with the SVC model, the Linear SVC has additional parameters such as penalty normalization which applies 'L1' or 'L2' and loss function. The kernel method can not be changed in linear SVC, because it is based on the kernel linear method.

6.1.5 Analyze and Prediction:

In the actual dataset, we chose only 2 features

1. **Review** : Review of a patient

2. **Labels** : Labels

Positive
Negative

6.1.6 Accuracy on test set:

We got a accuracy of 83.02% on test set.

6.1.7 Saving the Trained Model:

Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or . pkl file using a library like pickle .

Make sure you have pickle installed in your environment.

Next, let's import the module and dump the model into . pkl file

Database connecting using Mysql

So long as that works, do a quick control+d to exit the python instance.

Next, we want to make a Python file that can connect to the database. Generally you will have a separate “connect” file, outside of any main files you may have. This is usually true across languages, and here's why. Initially, you may have just a simple __init__.py, or app.py, or whatever, and that file does all of your operations. What can happen in time, however, is that your website does other things. For example, with one of my websites, Sentdex.com, I perform a lot of analysis, store that analysis to a database, and I also operate a website for users to use. Generally, for tasks, you will use what is called a “cron.” A cron is a

scheduled task that runs when you program it to run. Generally this runs another file, almost certain to not be your website's file. So then, to connect to a database, you'd have to write the database connecting code again in the file being run by your cron.

As time goes on, these sorts of needs stack up where you have some files modifying the database, but you still want the website to be able to access it, and maybe modify it too. Then, consider what might happen if you change your database password. You'd then need to go to every single file that connects to the database and change that too. So, usually, you will find the smartest thing to do is to just create one file, which houses the connection code.

Import the module.

Create a connection function to run our code. Here we specify where we're connecting to, the user, the user's password, and then the database that we want to connect to.

As a note, we use "localhost" as our host. This just means we'll use the same server that this code is running on. You can connect to databases remotely as well, which can be pretty neat. To do that, you would connect to a host by their IP, or their domain. To connect to a database remotely, you will need to first allow it from the remote database that will be accessed/modified.

Next, let's go ahead and edit our `__init__.py` file, adding a register function. For now we'll keep it simple, mostly just to test our connection functionality.

We allow for GET and POST, but aren't handling it just yet.

We're going to just try to run the imported connection function, which returns `c` and `conn` (cursor and connection objects).

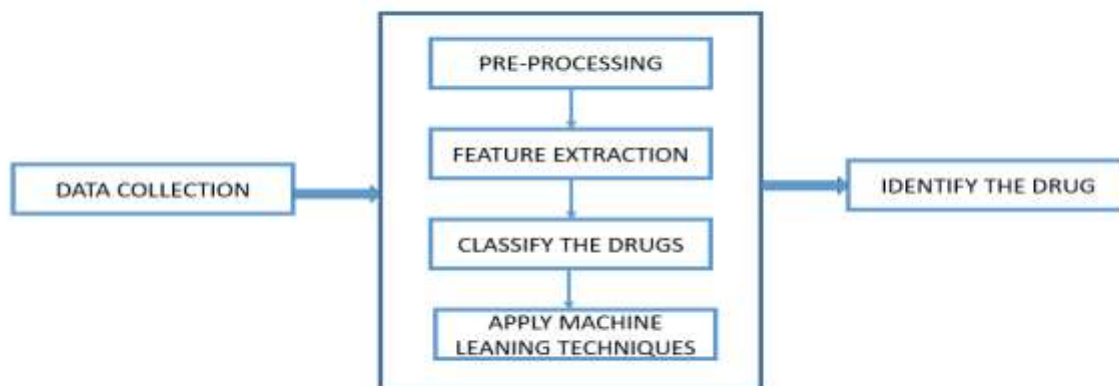
If the connection is successful, we just have the page say okay, otherwise it will output the error.

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 SYSTEM ARCHITECTURE

Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.



CHAPTER 8

SYSTEM TESTING

8.1 Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

8.2 Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

8.3 Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

8.4 Basics of software testing

There are two basics of software testing: black box testing and white box testing.

8.5 Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

8.6 White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

8.7 Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

8.7.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

8.7.2 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

8.7.3 Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

8.7.4 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

8.7.5 Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

8.7.6 Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

8.7.7 Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

8.7.8 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

8.7.9 Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and

is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

CHAPTER 9

CONCLUSION

Reviews are becoming an integral part of our daily lives; whether go for shopping, purchase something online or go to some restaurant, we first check the reviews to make the right decisions. Motivated by this, in this research sentiment analysis of drug reviews was studied to build a recommender system using different types of machine learning classifiers, such as Logistic Regression, Perceptron, Multinomial Naive Bayes, Ridge classifier, Stochastic gradient descent, LinearSVC, applied on Bow, TF-IDF, and classifiers such as Decision Tree, Random Forest, Lgbm, and Catboost were applied on Word2Vec and Manual features method. We evaluated them using five different metrics, precision, recall, f1score, accuracy, and AUC score, which reveal that the Linear SVC on TF-IDF outperforms all other models with 93% accuracy. On the other hand, the Decision tree classifier on Word2Vec showed the worst performance by achieving only 78% accuracy. We added best-predicted emotion values from each method, Perceptron on Bow (91%), LinearSVC on TF-IDF (93%), LGBM on Word2Vec (91%), Random Forest on manual features (88%), and multiply them by the normalized usefulCount to get the overall score of the drug by condition to build a recommender system.

Future Work:

Future work involves comparison of different oversampling techniques, using different values of n-grams, and optimization of algorithms to improve the performance of the recommender system.

Source code:

```
from flask import Flask, render_template, request, url_for, session, redirect,  
flash
```

```
from flask_mysqlldb import MySQL
```

```
import MySQLdb.cursors
```

```
import re
```

```
import pickle
```

```
import pandas as pd
```

```
import pandas as pd
```

```
import re
```

```
from flask import Flask, redirect, url_for, request, render_template
```

```
from werkzeug.utils import secure_filename
```

```
from keras.preprocessing.text import Tokenizer
```

```
from keras.preprocessing import sequence
```

```
from keras.models import Model, Input, Sequential, load_model
```

```
import pickle
```

```
import h5py
```

```
import numpy as np # linear algebra
```

```
import pandas as pd

from sklearn.feature_extraction.text import CountVectorizer

from keras.preprocessing.text import Tokenizer

from keras.preprocessing.sequence import pad_sequences

from keras.models import Sequential

from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D

from sklearn.model_selection import train_test_split

from keras.utils.np_utils import to_categorical

import re

import pickle

app = Flask(__name__)

# read object TfidfVectorizer and model from disk

drugs = pickle.load( open('drug.pkl', 'rb') )

tfids = pickle.load( open('tfid.pkl', 'rb') )

app.secret_key = 'neha'

app.config['MYSQL_HOST'] = 'localhost'

app.config['MYSQL_USER'] = 'root'
```

```
app.config['MYSQL_PASSWORD'] = 'root'
```

```
app.config['MYSQL_DB'] = 'drug'
```

```
mysql = MySQL(app)
```

```
@app.route('/', methods = ['GET','POST'])
```

```
def first():
```

```
    return render_template('first.html')
```

```
@app.route('/loginad')
```

```
def loginad():
```

```
    return render_template('loginad.html')
```

```
@app.route('/performance')
```

```
def performance():
```

```
    return render_template('performance.html')
```

```
@app.route('/upload')
```

```
def upload():
```

```
    return render_template('upload.html')
```

```
@app.route('/preview', methods=["POST"])
```

```
def preview():  
  
    if request.method == 'POST':  
  
        dataset = request.files['datasetfile']  
  
        df = pd.read_csv(dataset,encoding = 'unicode_escape')  
  
        df.set_index('Id', inplace=True)  
  
        return render_template("preview.html",df_view = df)
```

```
@app.route('/login', methods = ['GET', "POST"])
```

```
def login():  
  
    # Output message if something goes wrong...  
  
    msg = "  
  
    # Check if "username" and "password" POST requests exist (user  
submitted form)  
  
    if request.method == 'POST' and 'username' in request.form and  
'password' in request.form:  
  
        # Create variables for easy access  
  
        username = request.form['username']  
  
        password = request.form['password']  
  
        # Check if account exists using MySQL
```

```

cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

cursor.execute('SELECT * FROM people WHERE username = %s AND
password = %s', (username, password))

# Fetch one record and return result

account = cursor.fetchone()

# If account exists in accounts table in our database

if account:

    # Create session data, we can access this data in other routes

    session['loggedin'] = True

    global Id

    session['Id'] = account['Id']

    Id = session['Id']

    session['username'] = account['username']

    # Redirect to home page

    return redirect(url_for('index'))

else:

    # Account doesn't exist or username/password incorrect

    flash('Incorrect username/password! Please login with correct
credentials')

```

```

        return redirect(url_for('login'))

# Show the login form with message (if any)

return render_template('login.html', msg=msg)

@app.route('/register', methods= ['GET', "POST"])
def register():

    # Output message if something goes wrong...

    msg = ''

    # Check if "username", "password" and "email" POST requests exist (user
submitted form)

    if request.method == 'POST' and 'username' in request.form and
'password' in request.form and 'email' in request.form and 'age' in
request.form:

        # Create variables for easy access

        username = request.form['username']

        password = request.form['password']

        email = request.form['email']

        age = request.form['age']

```

```
reg      =      "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!#%*?&]{6,10}$"
```

```
pattern = re.compile(reg)
```

```
cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
```

```
# Check if account exists using MySQL)
```

```
cursor.execute('SELECT * FROM people WHERE Username = %s',  
(username,))
```

```
account = cursor.fetchone()
```

```
# If account exists show error and validation checks
```

```
if account:
```

```
    msg = 'Account already exists!'
```

```
elif not re.match(r'^[@]+@[^@]+\.[^@]+', email):
```

```
    msg = 'Invalid email address!'
```

```
elif not re.match(r'[A-Za-z0-9]+', username):
```

```
    msg = 'Username must contain only characters and numbers!'
```

```
elif not re.search(pattern,password):
```

```
    msg = 'Password should contain atleast one number, one lower case  
character, one uppercase character,one special symbol and must be between 6  
to 10 characters long'
```

```
elif not username or not password or not email:
```

```
    msg = 'Please fill out the form!'
```

else:

**# Account doesnt exists and the form data is valid, now insert new
account into employee table**

**cursor.execute('INSERT INTO people VALUES (NULL, %s, %s, %s,
%s)', (username, password, email, age))**

mysql.connection.commit()

flash('You have successfully registered! Please proceed for login!')

return redirect(url_for('login'))

elif request.method == 'POST':

Form is empty... (no POST data)

msg = 'Please fill out the form!'

return msg

Show registration form with message (if any)

return render_template('register.html', msg=msg)

@app.route('/index')

def index():

return render_template("index.html")

@app.route('/drug',methods=['POST','GET'])

def drug():

```
if request.method == 'POST':
```

```
    query_content=request.form['news_content']
```

```
    total= query_content
```

```
    total = re.sub('<[^>]*>', '', total)
```

```
    total = re.sub(r'[^\w\s]', '', total)
```

```
    total = total.lower()
```

```
    data=[total]
```

```
    vect=tfids.transform(data).toarray()
```

```
    prediction=drugs.predict(vect)
```

```
    pred=format(prediction[0])
```

```
login()
```

```
details = request.form
```

```
news_content = details['news_content']
```

```
drug_name = details['drug_name']
```

```
cur = mysql.connection.cursor()

cur.execute("INSERT INTO
review(news_content,pred,userid,drug_name) VALUES ( %s, %s,%s,%s) ",
(news_content,pred,Id,drug_name))
```

```
mysql.connection.commit()
```

```
cur.close()
```

```
return redirect('/users')
```

```
return render_template('index.html')
```

```
@app.route('/users')
```

```
def users():
```

```
cur = mysql.connection.cursor()
```

```
    resultValue = cur.execute(" SELECT * from people INNER JOIN review  
ON people.ID = review.USERID;")
```

```
    if resultValue > 0:
```

```
        userDetails = cur.fetchall()
```

```
    return render_template('users.html',userDetails=userDetails)
```

```
@app.route('/admin')
```

```
def admin():
```

```
    cur = mysql.connection.cursor()
```

```
    resultValue = cur.execute(" SELECT * from people INNER JOIN review  
ON people.ID = review.USERID;")
```

```
    if resultValue > 0:
```

```
        userDetails = cur.fetchall()
```

```
    return render_template('admin.html',userDetails=userDetails)
```

```
@app.route('/userdetail')
```

```
def userdetail():
```

```
    cur = mysql.connection.cursor()
```

```

cur.execute("SELECT * from people")

useradmin=cur.fetchall()

print(useradmin)


return render_template('userdetail.html',useradmin=useradmin)


@app.route('/chart')
def chart():

    legend = "review by drug_name"

    cursor = mysql.connection.cursor()

    try:

        cursor.execute("SELECT  drug_name  from  review  GROUP  BY
drug_name")

        # data = cursor.fetchone()

        rows1 = cursor.fetchall()

        labels = list()

        i = 0

        for row1 in rows1:

            labels.append(row1[i])

```

```

    cursor.execute("SELECT  drug_name  from  review  GROUP  BY
drug_name")

    # data = cursor.fetchone()

    rows2 = cursor.fetchall()

    label = list()

    j = 0

    values = list()

    k = 0

    for row2 in rows2:

        label.append(row2[j])

        cursor.execute("SELECT COUNT(id) from review WHERE  pred =
'negative' and drug_name=%s", (row2[j],))

        rows3 = cursor.fetchall()

        #j=j+1

    # Convert query to objects of key-value pairs

```

```

        for row3 in rows3:

            values.append(row3[k])

        #k=k+1

    mysql.connection.commit()

    cursor.close()

except:

    print('Error: unable to fetch items')


    return render_template('chart.html', values=values, labels = labels,
legend=legend)

@app.route('/worry')

def worry():

    legend = "review by drug_name"

    cursor = mysql.connection.cursor()

    try:

        cursor.execute("SELECT drug_name from review GROUP BY
drug_name")

        # data = cursor.fetchone()

        rows1 = cursor.fetchall()

        labels = list()

```

i = 0

for row1 in rows1:

labels.append(row1[i])

**cursor.execute("SELECT drug_name from review GROUP BY
drug_name")**

data = cursor.fetchone()

rows2 = cursor.fetchall()

label = list()

j = 0

values = list()

k = 0

for row2 in rows2:

label.append(row2[j])

**cursor.execute("SELECT COUNT(id) from review WHERE pred =
'positive' and drug_name=%s", (row2[j],))**

rows3 = cursor.fetchall()

```

        #j=j+1

# Convert query to objects of key-value pairs


        for row3 in rows3:

            values.append(row3[k])

        #k=k+1

mysql.connection.commit()

cursor.close()


except:

    print('Error: unable to fetch items')


    return render_template('worry.html', values=values, labels = labels,
legend=legend)


@app.route('/negative')

def negative():

```

```

cur = mysql.connection.cursor()

cur.execute("SELECT * from people INNER JOIN review ON people.ID =
review.USERID WHERE pred ='Negative'")

useradmins=cur.fetchall()

print(useradmins)

return render_template('negative.html',useradmins=useradmins)

@app.route('/postive')

def postive():

cur = mysql.connection.cursor()

cur.execute("SELECT * from people INNER JOIN review ON people.ID =
review.USERID WHERE pred ='Positive'")

useradmi=cur.fetchall()

print(useradmi)

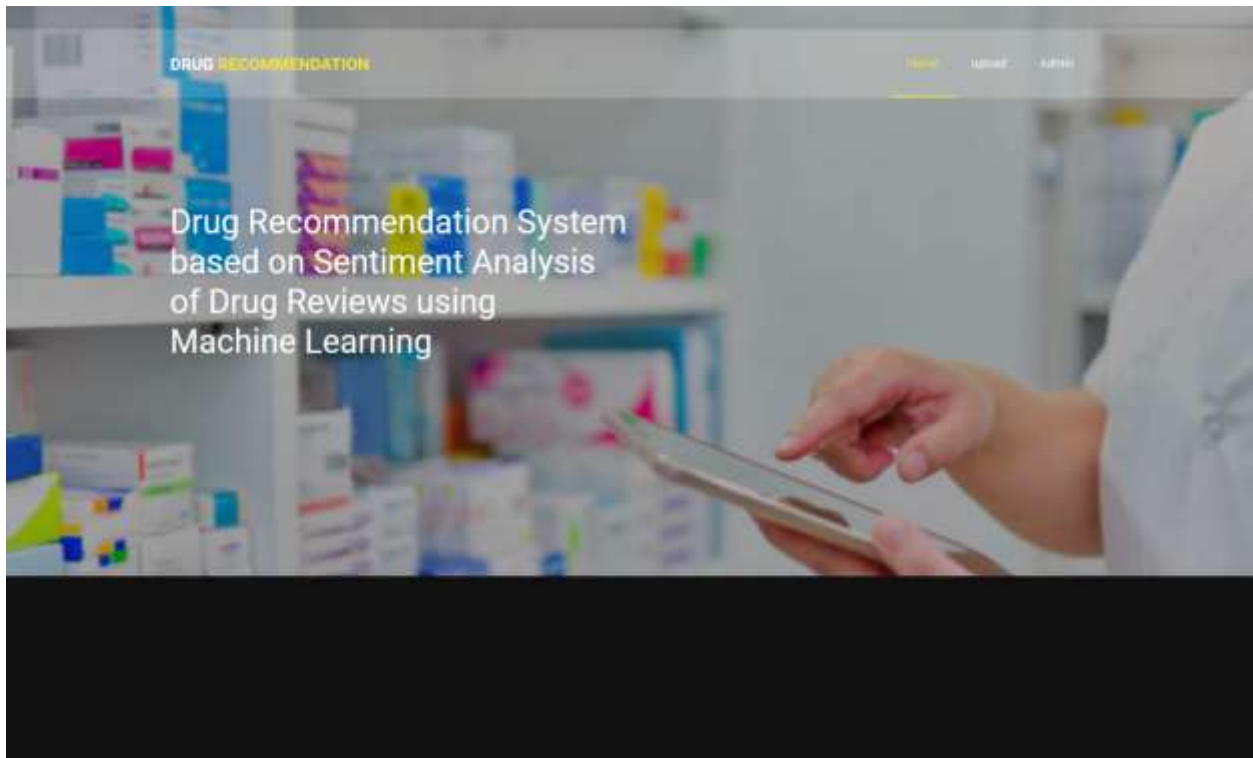
return render_template('postive.html',useradmi=useradmi)

```

```
if __name__ == '__main__':
```

```
    app.run()
```

SCREENSHOTS



upload

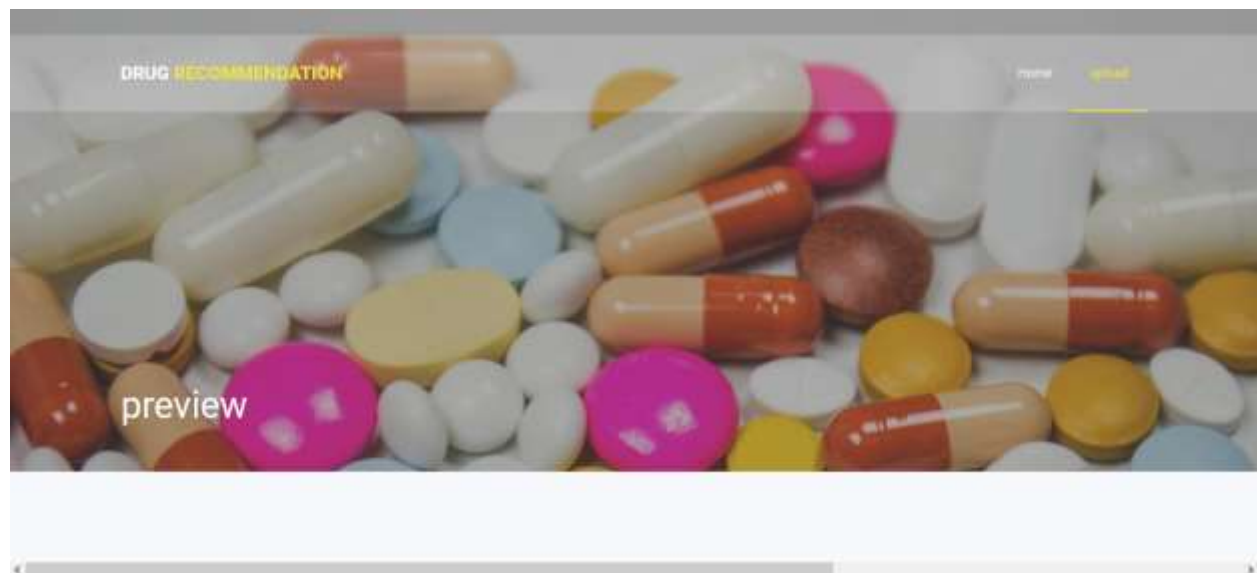
Upload

No file selected

upload

Upload

upload.docx



preview

Preview

drugName	condition	
mirtazapine	Depression	"I've tried a few antidepressants over the years (citalopram, fluoxetine, amitriptyline), but none of those helped with my depression. My doctor suggested and changed me onto 45mg mirtazapine and this medicine has saved my life. Thankfully I have had no side effects, no weight gain, I've actually lost a lot of weight. I still have suicidal thoughts but mirtazapine has saved me."
aspirin	Crohn's Disease Maintenance	"My son has Crohn's disease and has done very well on the Aspirin. He has no complaints and shows no side effects. He has

asone / terol	Asthma	"I've had asthma since I was 5 and now I'm 42. I've always only used rescue inhalers for most of it and since I've my inhaler and I'm never short of breath. It's apparent that this is not for everyone as I've read some reviews. I have never had any side effects. I'm a fan... actually was given a free 12 month prescription for this and that alone saved m guessing!"
alopram	Anxiety	"If you just started Lexapro and are considering quitting because of the side effects - don't!! Trust me. Wait it out. It gets so n Lexapro, I was extremely nauseous and anxious, and couldn't sleep at night. I've been on it now for six months and fe The only side effect I experience is occasional drowsiness. I can sleep at night, I have no sexual side effects, and I haven't g off this medicine without consulting your Doctor first and having him/her help you through it. If you take yourself off it (especially c Be smart and stay strong. The good definitely outweighs the bad."
termine	Weight Loss	"Started Phentermine 37.5mg about 10 days ago. Weight 196lb and 5'8". MD suggested due to health problems and impair only took the pill 2 days; skip a day, one pill, and skip a day again. I could barely eat bites of food... that's why I staggered and felt "off", so I'd force myself to eat fruit, or something small. I really want the weight off but I know diet pill... just short term booster. I do exercise 3 x week light low impact. I plan to work hard with healthy eating, exercise, and this g for the assistance, but realistically learning to have healthy habits is key!"
inum chloride hydrate	Hyperhidrosis	"I have always had excessive sweating and body odour because of that and always been looking for medication for this. Tried so m they did not seem to work at all. I put this on for 2 nights continuously and I was surprised at the results! It really works! Although it i it's worth the price."

Provera	Abnormal Uterine Bleeding	"I am a user of the birth control shot and I honestly love the fact that I do not have to worry about taking the pill. I just go to the doc. But the only thing I hate is the bleeding for long periods. I have been on the shot for almost a year and a half. I remember the first 12 pain one day in my stomach, it felt so bad. I was in class and I had to run to the bathroom. When I was sitting on the toilet, this HUG the size of a tennis ball. Afterwards, my cramping stopped."
sprintec	Birth Control	"Tru sprintec makes me sick every single day, like throwing up constantly unless I eat a lot. That is the downside. Upside: It cleared, lose weight, overall a good medicine for acne, maybe not for a birth control?"

[CLICK TO TRAIN | TEST](#)

Register

Register

submit

Register

Register

and

and@gmail.com

12

SUBMIT

DRUG RECOMMENDATION

[home](#)[upload](#)[login](#)[register](#)

Login

Login

Username :

Password :

login

DRUG RECOMMENDATION

[Home](#)[upload](#)[login](#)[register](#)

Login

Login

Username :

anul

Password :

login

Drug Reviews

Drug Reviews using Machine Learning

Enter your Review

drug_name: Sulfamethoxazole / trimethoprim

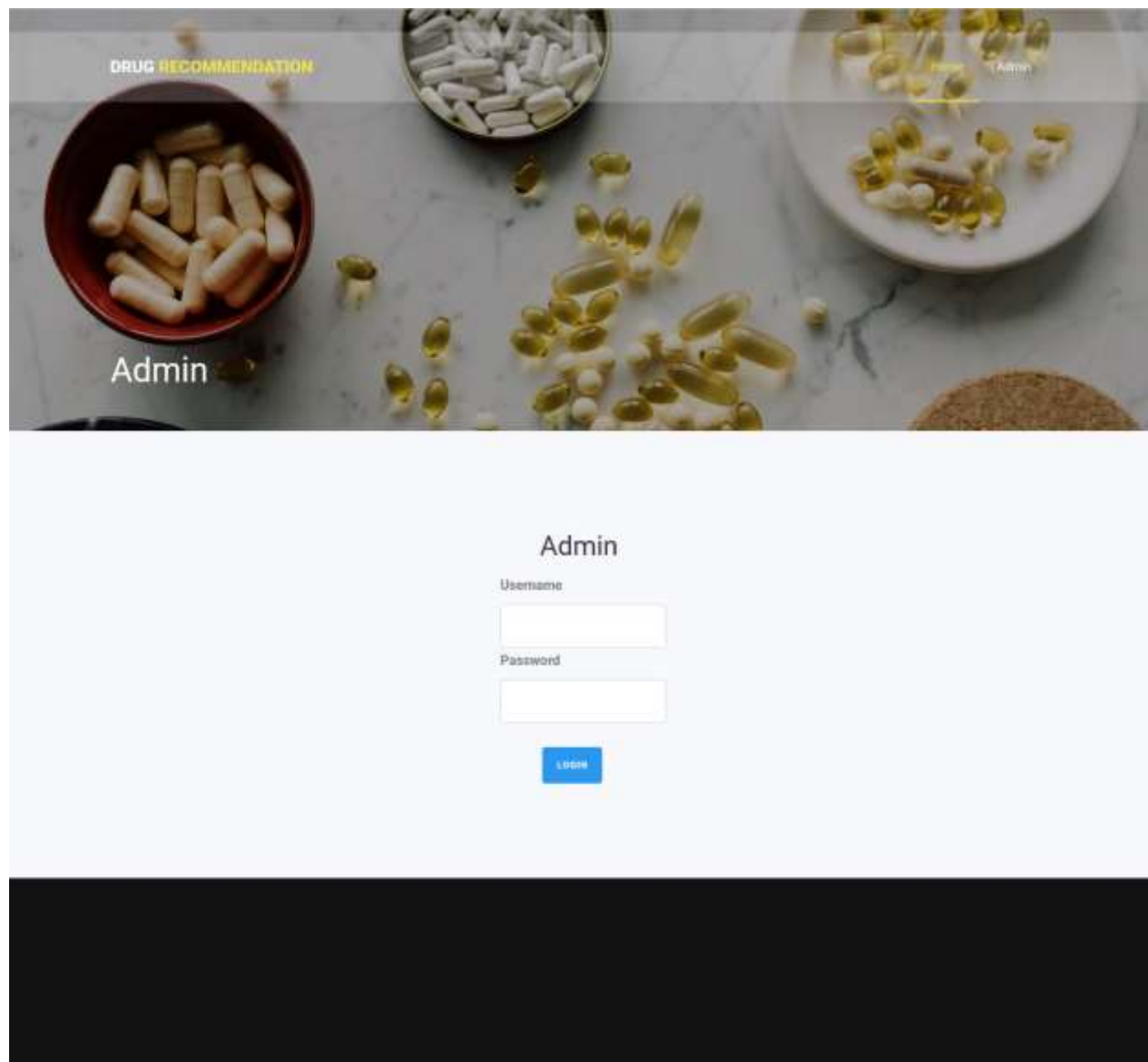
"Was prescribed one dose over the course of one day, took 4 pills of 250mg after a light lunch, and had nausea and mild stomach pains/upset. Lying down did not alleviate the

SUBMIT

Review details

Drug Review details

User ID	Username	Email_Id	age	Review	post
1	santhosh	Sandy@123	19	"I had a wonderful experience with Ziana. This is the only acne medication that didn't dry out my skin after using it. I thought I would never have a good experience with acne medications. Ziana is a smooth, moisturizing acne medication that leaves your skin feeling great. With a couple days the redness and swelling of my acne went down significantly. I don't see myself ever using another acne medication."	Your review Has Been Posted Successfully
1	santhosh	Sandy@123	19	"Taking drug for about 5 years and blood pressure stays around 140/93. All of a sudden started having ringing in ears and decided to go off Diovan after reading this was a side effect. Also noticed shortness of breath, weight gain and tiredness. Now blood pressure is about the same without Diovan or any other medication except still have the ringing in the ears."	Your review Has Been Posted Successfully
2	arui	Sandy@123	12	"Was prescribed one dose over the course of one day, took 4 pills of 250mg after a light lunch, and had nausea and mild stomach pains/upset. Lying down did not alleviate the	Your review Has Been Posted Successfully



Admin

Admin

Username

admin

Password

Login



Register details

Register details

user_id	user_name	Email	age
1	santhosh	Sandy@123	18
2	arul	Sandy@123	12

Review details

Review details

user_id	user_name	Email	age	drug Name	Review	pred
1	santhosh	Sandy@123	18	Ziana	"I had a wonderful experience with Ziana. This is the only acne medication that didn't dry out my skin after using it. I thought I would never have a good experience with acne medications. Ziana is a smooth, moisturizing acne medication that leaves your skin feeling great. With a couple days the redness and swelling of my acne went down significantly. I don't see myself ever using another acne medication."	Positive
1	santhosh	Sandy@123	18	Diovan	"Taking drug for about 5 years and blood pressure stays around 140/93. All of a sudden started having ringing in ears and decided to go off Diovan after reading this was a side effect. Also noticed shortness of breath, weight gain and tiredness. Now blood pressure is about the same without Diovan or any other medication except still have the ringing in the ears."	Negative
2	arul	Sandy@123	12	Sulfamethoxazole / trimethoprim	"Was prescribed one dose over the course of one day, took 4 pills of 250mg after a light lunch, and had nausea and mild stomach pains/upset. Lying down did not alleviate the	Positive

Positive Review details

Positive Review details

user_id	user_name	Email	age	drug Name	Review	pred
1	santhosh	Sandy@123	18	Ziana	"I had a wonderful experience with Ziana. This is the only acne medication that didn't dry out my skin after using it. I thought I would never have a good experience with acne medications. Ziana is a smooth, moisturizing acne medication that leaves your skin feeling great. With a couple days the redness and swelling of my acne went down significantly. I don't see myself ever using another acne medication."	Positive
2	arul	Sandy@123	12	Sulfamethoxazole / trimethoprim	"Was prescribed one dose over the course of one day, took 4 pills of 250mg after a light lunch, and had nausea and mild stomach pains/upset. Lying down did not alleviate the	Positive

Negative Review details

Negative Review details

user_id	user_name	Email	age	drug Name	Review	pred
1	santhosh	Sandy@123	18	Diovan	"Taking drug for about 5 years and blood pressure stays around 140/93. All of a sudden started having ringing in ears and decided to go off Diovan after reading this was a side effect. Also noticed shortness of breath, weight gain and tiredness. Now blood pressure is about the same without Diovan or any other medication except still have the ringing in the ears."	Negative

Performance Analysis

PERFORMANCE ANALYSIS

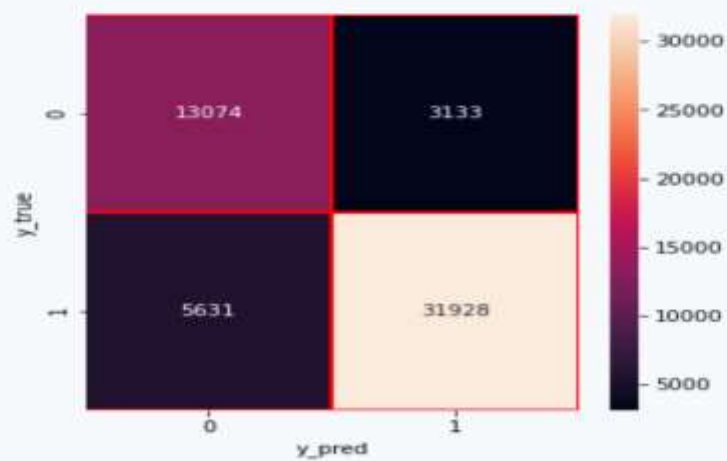
Precision and recall

Recall Precision

Negative	0.70 0.81
----------	-----------

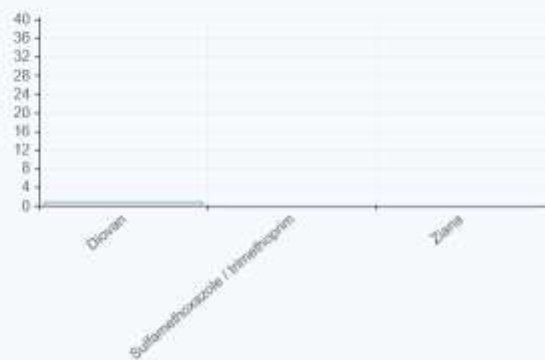
Positive	0.91 0.85
----------	-----------

Confusion Matrix



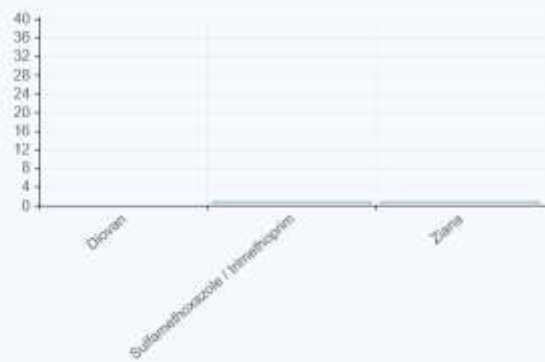
chart

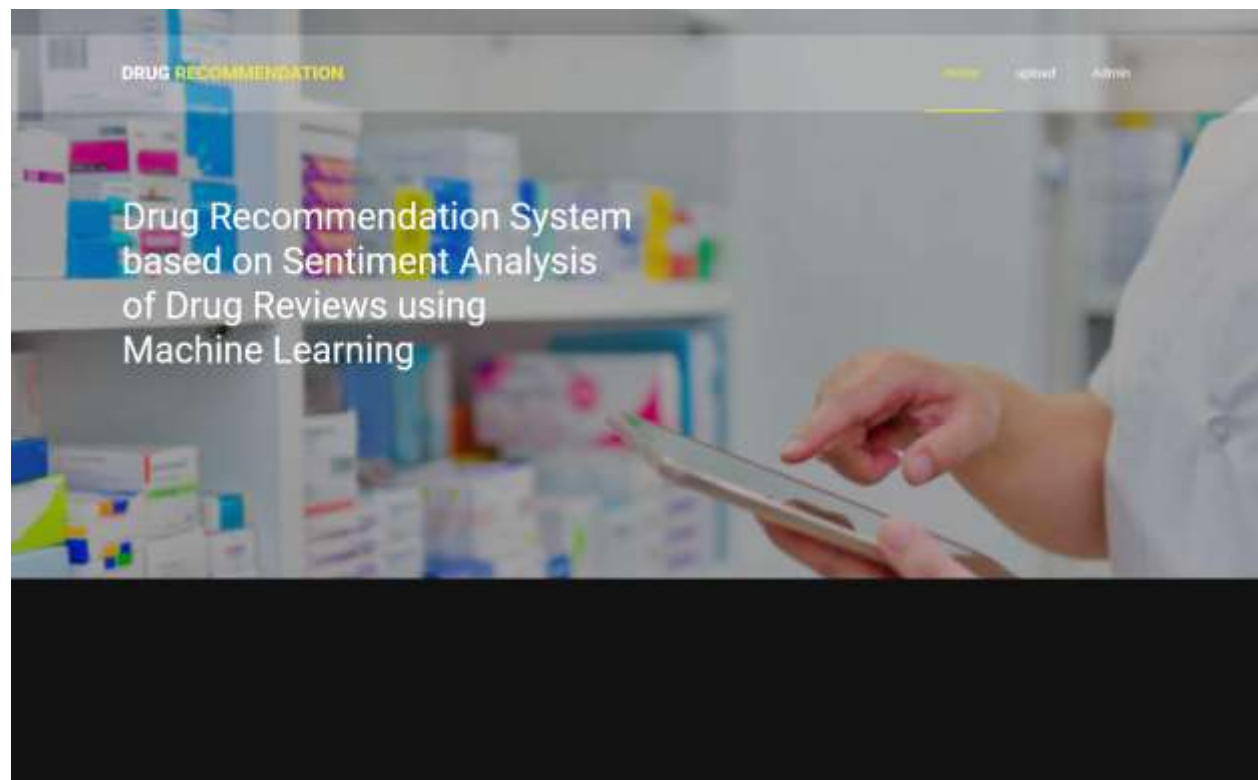
Analysis on Drug Reviews



chart

Analysis on Drug Reviews





REFERENCES

- [1] Chengquan Zhang et al., "Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes", *arXiv preprint arXiv:1904.06535*, 2019.
- [2] Y. Wu, X. Yang, J. Bian, Y. Guo, H. Xu and W. Hogan, "Combine Factual Medical Knowledge and Distributed Word Representation to Improve Clinical Named Entity Recognition", *AMIA*, 2018.
- [3] Shi Baoguang, Xiang Bai and Serge Belongie, "Detecting oriented text in natural images by linking segments", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [4] A. Stubbs, M. Filannino and Ö. Uzuner, "De-identification of psychiatric intake records: Overview of 2016 CEGS N-GRID shared tasks Track 1", *J. Biomed. Inform.*, vol. 75S, pp. S4-S18, Nov. 2017.
- [5] T. Mikolov, E. Grave, P. Bojanowski, C. Puhersch and A. Joulin, "Advances in Pre-Training Distributed Word Representations", *CoRR*, vol. abs/1712.09405, 2017.
- [6] A. E. W. Johnson et al., "MIMICIII a freely accessible critical care database", *Sci. Data*, vol. 3, pp. 160035, May 2016.
- [7] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou and T. Mikolov, "FastText.zip: Compressing text classification models", *Arxiv Prepr. ArXiv161203651*, 2016.