# INTRODUCTION

## PROJECT INTRODUCTION

With the number of coronavirus 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 Telemedicine framework ought to be energized as far as possible in this difficult time . 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. Choosing the toplevel medication is significant for patients who need specialists that know wide-based information about microscopic organisms, antibacterial medications, and patients . 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. With the exponential development of the web and the web-based business industry, item reviews have become an imperative and integral factor for acquiring items worldwide. Individuals worldwide become adjusted to analyze reviews and websites first before settling on a choice to buy a thing. While most of past exploration zeroed in on rating expectation and proposals on the E-Commerce field, the territory of medical care or clinical therapies has been infrequently taken care of. There has been an expansion in the number of individuals worried about their well-being and finding a diagnosis online. As demonstrated in a Pew American Research center survey directed in 2013 , roughly 60% of grown-ups searched online for health-related subjects, and around 35% of users looked for diagnosing health conditions on the web. A medication recommender 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 recommender 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 recommender system, medicine is offered on a specific condition

## SCOPE

The Earth is passing through a purplish patch of technology, where there is increasing demand of intelligence and accuracy behind it. Today’s people are more likely addicted to Internet but they are not concerned about their personal health. In this 21st Century humans are surrounded with technology as they are the constituent of our day to day life cycle.So,We are proposing such a system which will flaunt a simple and elegant User Interface and also be time efficient. In order to make it less time consuming we are aiming at a more specific questionnaire which will be followed by the system. Our aim with this system is to be the connecting bridge between doctors and patients. The main feature will be the machine learning, in which we will be using algorithms such as Naïve Bayes Algorithm, K-Nearest Algorithm, Decision Tree Algorithm, Random Forest Algorithm and Support Vector Machine, which will help us in getting accurate predictions and Also, will find which algorithm gives a faster and efficient result by comparatively-comparing. Another feature that our system will comprise of is Doctor’s Consultation. After delivering the results, our system will also suggest the user to get a doctors consultation on this report.By using this feature, we will not only address the other class of users i.e. the Doctors but we will also gain their trust in this system as in that this system is not affecting their business.

## PROJECT OVERVIEW

Drug and Disease features were obtained by querying open linked data to train our classifier for predicting new drug indications, and the predictive performance of the classifier for different validation schemes was evaluated. We collected the drug and disease data from Bio2RDF, an open source project that uses semantic web technologies to link data from multiple sources. A binary feature matrix was generated using drug target, substructure and side effects and disease ontology terms. We collected a broader collection of data containing 816 drugs and 1393 diseases with their features and gold standard data we generated by combining multiple drug indication

data sources. We tried our method on a different dataset, compiled by other researchers, that confirmed the predictive value of our method independent of the primary data. A crucial flaw in the typical evaluation scheme for drug indication predictions that would yield unrealistic predictions is to fail to consider the paired nature of inputs. We partitioned the data in distinct training and test sets where not only pairs but also drugs/diseases are were not overlapped. We tested several classifiers.

## OBJECTIVES

The system is more effective since it presents the proposed algorithm used in natural language processing responsible for counting the number of times of all the tokens in review or document. The system has exact sentiment analysis prediction techniques for Data Cleaning and Visualization

#### The following main project goals are :

1. The right patient.
2. The right drug
3. The right dose

## DATASETS

We used Drug Review Dataset in two files (test and train) directly downloaded from Blob Storage in Azure Cloud. The Dataset contains a unique ID, drug name, condition, review, rating, date and user count. Then, we convert the ‘Rating’ and ‘Useful Count’ into a Numeric data type for generating a new column of ‘most reviewed’ drugs. The top 10 Most Reviewed Drugs is shown in Fig 2 where we can analyze the number of reviews for the 10 drugs. The graphical representation for the total number of drugs per condition (Top 20) and the number of reviews in each year is illustrated in Fig 3 and4 respectively. Figure 5 demonstrates the d by highest monthly reviews for different drugs.

#### Stemming using NLTK Library

Before going to this step, a new data frame was created with attributes like unique ID, drug name, condition and tags. Column ‘tags’ consists of concatenated values of a drug name, condition and reviews. We perform Stemming on the new data frame. Stemming

using NLTK Library is a technique used to extract the base form of the words by removing affixes from them. Porter Stemmer is applied to check root words. Word Net Lemmatizer is also deployed to check root words and meaning consistency.

#### Implementing Count Vectorizer to make words in vector form

In this step, column ‘tags’ is considered. Count vectorizer is used to make the words in vector form. Maximum features is set to 500 for removing a d stop words from a string. Applied Fit transform to Lemma processed vector as it gives more relevant meaning.

The UCF101 dataset comprises a wide range of action categories, including activities such as applying eye makeup, applying lipstick, baby crawling, balance beam, band marching, bench press, basketball shooting, basketball slam, biking, billiards shot, and blow drying hair. Additionally, the dataset includes action categories like Boxing Punching Bag, Boxing Speed Bag, Breaststroke, Brushing Teeth, and Bowling. It also encompasses action categories such as cliff diving, Jerk and Clean, cricket, cricket shot, kitchen chopping, drumming, fencing, cricket Floor Gymnastics, Frisbee Catch, Front Crawl, Golf Swing, Head Massage, Hammer Throw, Hammering, Handstand Pushups, Handstand Walking, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jacks, Kayaking, Knitting, Long Jump, and Lunges, a military parade Pizza Tossing, Nun Chucks, Parallel Bars, and Batter Mixing while playing the piano, guitar, tabla, and so on...

#### Logistic Regression

Logistic Regression In this step, to understand whether the dataset is a useful dataset and powerful, logistic regression is applied to identify the new recommendation from this dataset is valid or not. Basically, it checks the accuracy of the dataset. First, we import the Logistic Regression module and create a classifier object using the function Logistic Regression(). Then, using fit() function, we fit the model on train dataset. After that using predict() function, we perform prediction on test dataset. Then, we use confusion matrix for model evaluation. A confusion matrix is a table that is used for the evaluation of performance of a classification model. The model was evaluated using model evaluation metrics such as accuracy, precision, F1 score, and recall. Compiled

all of the metric into a data frame in Fig 6. The figure shows that it has a classification rate of 80% which is considered as good accuracy.

#### Evaluating Model with Various Similarity and Distance Metrics

Evaluating Model with Various Similarity and Distance Metrics In this step, we evaluated the model with various Similarity and Distance Metrics like Pearson Correlation, Spearman Correlation, Kendall Tau’s Correlation, Cosine Similarity, Jaccard’s Similarity, Euclidean distance and Manhattan Distance with first drug and random drugs.

#### Building Recommendation System

There are various approaches to build a recommendation system. We opted with two approaches following :

**Content based filtering:** A content-based recommender works with information that is provided by the user, either explicitly (rating) or implicitly (clicking on a link). In this method, we by choosing cosine similarity for recommender system building. Then, sorting similarities in descending orders to make more similar drugs on top. Defined a function to recommend a drug based on the condition selected.

**Collaborative-based filtering**: In collaborative filtering, it finds similar users and recommend what similar users like. It recommends the items based on the reviews of the previous users. In this method, we are comparing for similar condition by building User-item interactions Matrix and compare for similar drug. User-item matrix is a basic foundation of traditional collaborative filtering techniques. We sorted the values according to the rating. Testing the drug matrix which will indicate the medicine which is suitable for the input condition and analysed correlation with similar condition.

# LITERATURE SURVEY

## EXISTING SYSTEM

Most of the people tend to live a long and healthy life, where they are more conscious about their health. But many studies show that almost many people die due to the medical errors caused in terms of taking wrong medicines and these errors are caused by doctors, who prescribe medicines based on their experiences which are quite limited. As machine learning, deeplearning and data mining like technologies that are emerging day by day, these technologies can help us to explore the medical history and can reduce medical errors by being doctor friendly

## PROPOSED SYSTEM

A recommender 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 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. On the other hand, Featuring engineering is the process of making more features from the existing ones; it improves the performance of models.

## RELATEDWORK

These days, recommender 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 . The study presents GalenOWL, a semanticempowered 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 GalenOWL, 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.

Leilei Sun 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 , 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 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. examined three distinct algorithms, decision tree algorithm, support vector machine (SVM), and backpropagation 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. developed a cloudassisted drug proposal (CADRE). As per patients’ side effects, CADRE can suggest drugs with top-N related prescriptions. This proposed framework was initially founded on collaborative filtering

techniques in which the medications are initially bunched into clusters as indicated by the functional description data. However, after considering its weaknesses like computationally costly, cold start, and information sparsity, the model is shifted to a cloud-helped approach using tensor decomposition for advancing the quality of experience of medication suggestion. Considering the significance of hashtags in sentiment analysis,

Jiugang Li et al. constructed a hashtag recommender framework that utilizes the skipgram model and applied convolutional neural networks (CNN) to learn semantic sentence vectors. These vectors use the features to classify hashtags using LSTM RNN. Results depict that this model beats the conventional models like SVM, Standard RNN. This exploration depends on the fact that it was undergoing regular AI methods like SVM and collaborative filtering techniques; the semantic features get lost, which has a vital influence in getting a decent expectation.

# SYSTEM ANALYSIS

## FUNCTION REQUIREMENTS

* Collect the dataset
* Load the data
* Train,Test split

The functional specifications ought to be comprehensive and consistent. The definition of completeness implies the definition of all user requirements. Consistency requires that every demand is explicitly defined without any ambiguous language. Due to issues that develop while specifying the functional requirements of these systems, it is generally recognised that completeness and consistency cannot be attained in large software or in a complicated system. Completeness and consistency cannot be achieved due to the various stakeholder needs. These factors could make requirements less evident when they are first described and could also result in inconsistent requirements specifications.

#### Collect The Dataset:

The largest dataset of human actions at the moment.comprises of authentic user- uploaded films, the majority of which are collected from random YouTube, and is divided 101 action classes, dispersed throughout 13k+ movies and 27 hours of video data, have been organised.

The most challenging action dataset, which contains the greatest number of classes, clips, and unconstrained clips, aims to further action recognition research by identifying and experimenting with new realistic action categories.

#### Load the data first :

Get the datasets loaded: Handling Datasets Data Visualization Train, Test Split We will stack the informational collection of disdainful discourse that is expected to be ordered by the cannot stand discourse.

#### Data Visualization:

After that, data visualization comes next.

#### The split between training and testing :

The intuitively end users may engage with the system, emphasising its simplicity and ease of use In order to identify disparaging speech and establish whether it belongs to a group or not, we will build a deep learning model in which the results are produced by combining the classifiers with the highest accuracy

Some of the non functional requtrements are:

The system's speed, scalability, and responsiveness are all performance requirements. These requirements include the system's capacity to manage a specific number of concurrent users and process a specified volume of data efficiently within set time restrictions.

Security criteria are concerned with protecting the system and its data from unauthorised access while also assuring the capacity to discover and correct any possible security issues.

Usability considerations focus around how easily and.

Reliability refers to the system's ability to work appropriately and consistently in both expected and unforeseen situations.

Maintainability refers to the requirements that must be satisfied in order for the system to be easy to test, debug, and modify.

Portability refers to a system's ability to be readily moved to different hardware or software environments.

Compliance: The actions required to guarantee that policies, regulations, & laws are followed.

Non-functional requirements are critical because they can have a significant influence on the overall success and quality of a software system. The performance, security, and

usability of the system are typically closely related to them. Additionally, they assist in ensuring that the system is maintainable, portable, and compliant with all relevant regulations and laws.

## PERFORMANCE REQUIREMENTS

* The effectiveness of a software system is described by its performance requirements. Response time, execution time, storage capacity, and throughput are all aspects of a program's performance. The majority of the service level performance criteria were created to support end-user tasks. Performance requirements are crucial to the design and testing of software, just like other quality criteria are. Despite being an important step in the development of software, acquiring requirements can be challenging. The three key challenges in gathering performance criteria are scope, skill, and stability. Project scope is frequently ambiguous in the software development sector, because different stakeholders provide contradictory or inaccurate scope evaluations.

## SOFTWARE REQUIREMENTS

* System Pentium 13 processor.
* Hard disk 500 gb. Monitor: 15" led
* Input devices: keyboard, mouse Ram : 4GB

## HARDWARE REQUIREMENTS

* Operating system: windows 10.
* Coding language: python 3.8
* Web framework: flask

## FEASIBILITY STUDY

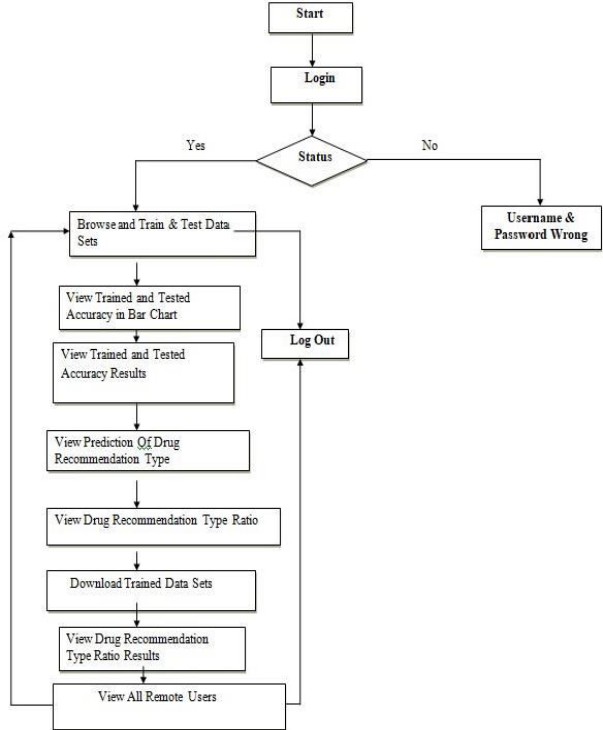
Leilei Sun examined large scale treatment records to locate the best treatment prescription for patients. The idea was to use an efficient semantic clustering algorithm estimating the similarities between treatment records. Likewise, the author created a framework to assess the adequacy of the suggested 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.

# SYSTEM DESIGN

Systems design is the process of def ning the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.

Object-oriented analysis and design methods are becoming the most widely usemethods for computer systems design.

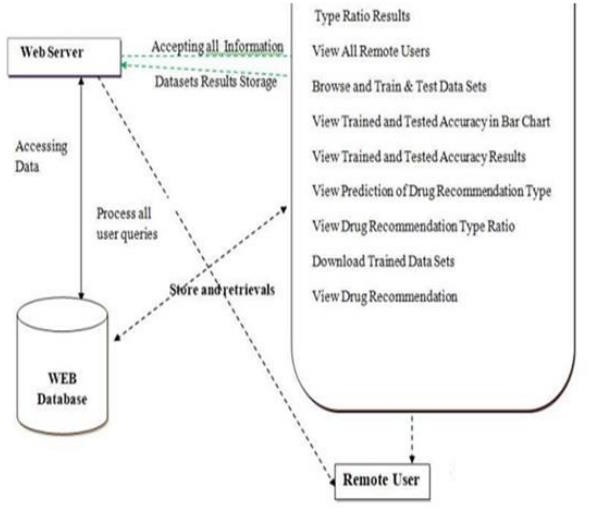
## SYSTEM ARCHITECTURE



#### FIGURE 4.1: System Architecture

A system architecture is the conceptual model that def nes the structure, behavior, and more views of a system.An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

## DATA FLOW DIAGRAM



#### FIGURE 4.2: Flowchart of data

The camera first identifies the presence of a human subject, then determines the exact activity being done. To do this, a deep learning model known as ResNet is used, which commonly employs 2D kernels. However, the model is augmented with the incorporation of 3D kernels to improve the efficacy of activity identification. The training of the activity recognition model is done with the kinetics UCF dataset, which contains over 3000 unique activity categories. Videos demonstrating each exercise are included to improve the model's accuracy.

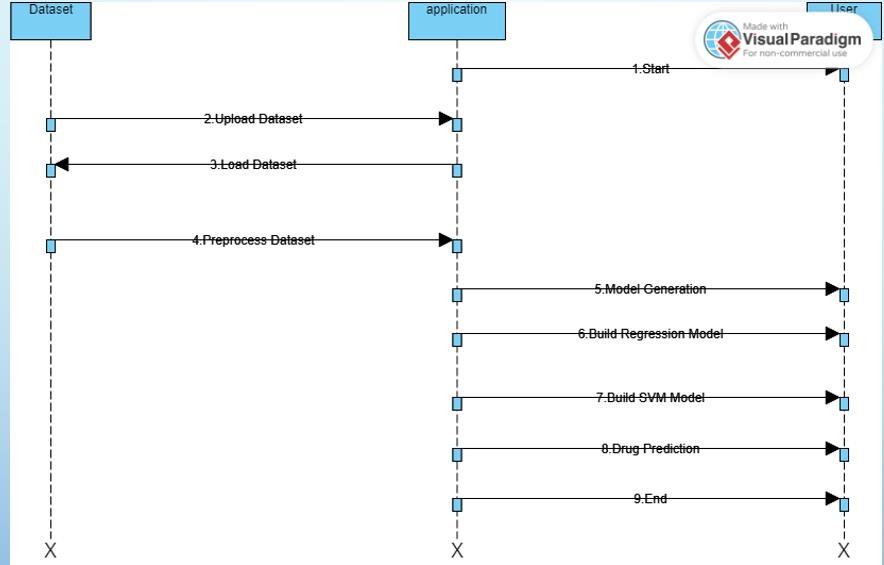
The ResNet model is used in conjunction with a 2D CNN (Convolutional Neural Network) architecture to recognise human activities. While the classic 2D CNN model is commonly used for image-based tasks such as detection, segmentation, and captioning, our approach enhances its capabilities by including a 3D CNN component. This supplementary 3D CNN analyses movies, which are a series of frames, as opposed to individual pictures in the 2D CNN. Incorporating these layers within the model enhances the chance of successful action recognition. The model greatly enhances the accuracy of human activity recognition by conducting tasks such as detection, summarization, optical flow analysis, segmentation, and captioning.

## UML DIAGRAMS

Diagrams in the UML can be classified into two different sorts. The majority of types describe common patterns of behaviour, including a handful that represent various facets of interactions, while some types represent structural information.

Structure diagrams depict the static components of a system. It emphasises the system needs being simulated. Because they depict the structure, structure diagrams are widely used to document the software architecture of software systems.Behaviour diagrams depict the system's dynamic nature. It emphasises the critical aspects of the system being simulated. Because they depict how a system operates, behaviour diagrams are commonly used to describe the functioning of software systems. The activity diagram, for example, goes into great depth on operational and business activities. UML Diagrams for our project are as follows:

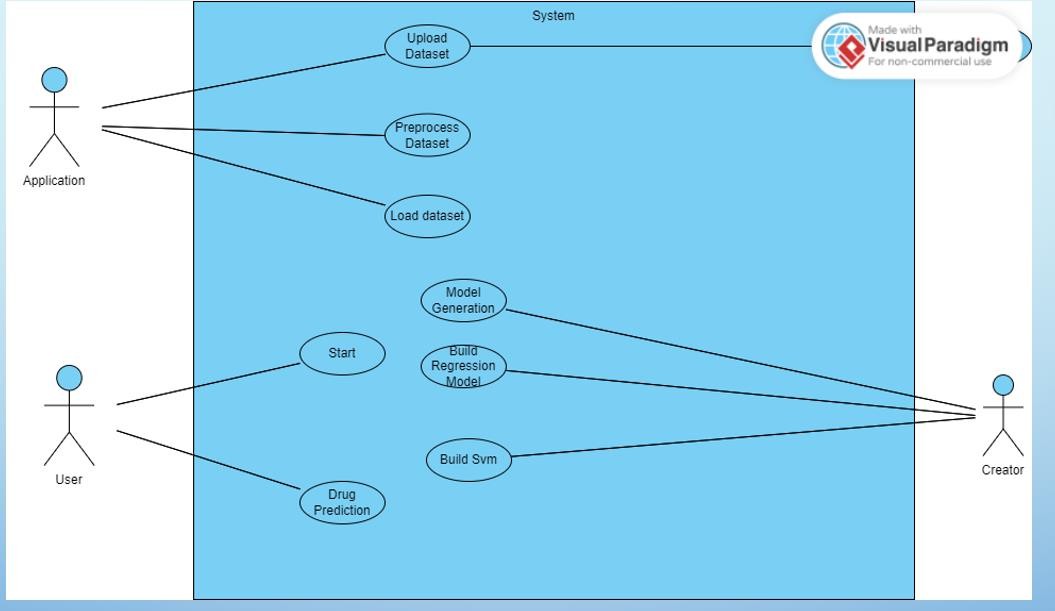
### Sequence Diagram



#### Figure 4.3: Sequence Diagram

A sequence diagram is a UML diagram that illustrates the flow of messages exchanged between objects or components in a system. It depicts the flow of communication and the sequence of interactions, assisting in the modelling of a system's dynamic behaviour.Sequence diagrams are commonly used for visualizing the interactions during runtime, capturing the timing and sequence of messages, and depicting the behaviour of objects or components in a system. They are widely used in software development for designing, documenting, and analyzing the interactions between entities in a system.

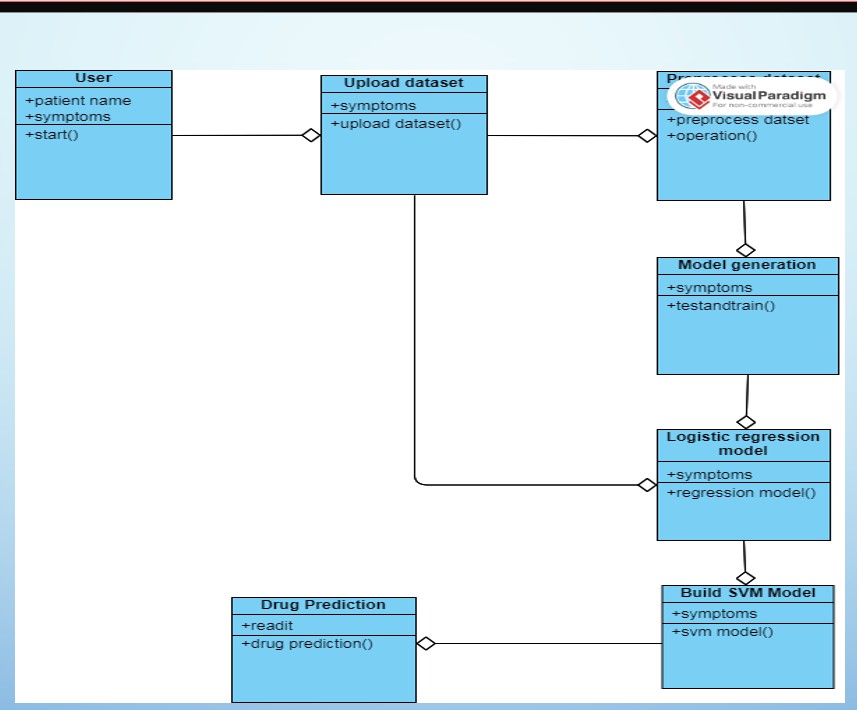
### Use Case Diagram



#### Figure 4.4: Use Case Diagram

The capacity to create use case diagrams is a key component of the Unified Modelling Language (UML). Use cases are used to categorise and identify system functions throughout a project's analysis phase. Actors and use cases are separated in the system. The various roles that system users can take on are represented by actors. The functionality of the fashion products picture classification is shown in this use case diagram. It has six different functions. Users are the actors in this scenario, and the use cases are Enter input, upload a fashion image, and features are extracted before a predicted output label, classification algorithm, and datasets are generated. Actor, or the user, must upload the image to the system, after which the ML model uses a variety of methods to forecast the image as depicted.Actor i.e. user has to upload the image to the system then it is taken by ml model to predict the image using various algorithms as shown in the use case diagram above.

### Class Diagram



#### Figure 4.5: Class Diagram

The class diagram is a static representation of an application that depicts the many sorts of objects in the system and their interactions. It displays a visual depiction of classes, their connections, attributes, and methods, providing a high-level perspective of the software system. A class can inherit from other classes in addition to creating its own objects, allowing for code reuse and supporting modular architecture.

Class diagrams are used in software development for a variety of reasons, including representation, definition, and code generation for a wide range of system components. They collect class names, characteristics, and methods and provide as a useful resource during the software development process. The class diagram acts as a structural diagram by providing classes, interfaces, affiliations, collaborations, and constraints, encapsulating the fundamental parts of the system's structure and behaviour.

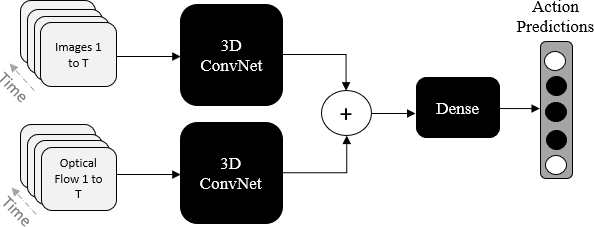
## MODULES

#### CNN :

A deep learning approach called a convolutional neural network (CNN) is used in applications for image processing and classification. It is a kind of neural network that begins by extracting data from pictures using a number of convolutional layers before using pooling layers to make the features' dimensions smaller. One or more completely linked layers receive the final output and use it to forecast future events. The fundamental benefit of CNNs over conventional image processing techniques is their capacity to automatically learn new features without the need for manual feature engineering. As a result, they excel at tasks like categorization, object identification, and picture segmentation. Existing CNN designs may be used with popular deep learning frameworks like TensorFlow and PyTorch.

#### I3D :

One of the fundamental contrasts between information included in a single image and information contained in a movie is the transitory component. As a result, deep learning model designs have been enhanced to include 3D processing to analyse temporal data further. In this article, the I3D model is used to describe the architectural changes from pictures to video.



**Figure 4.6: I3D Architecture**

# IMPLEMENTATION AND RESULTS

## LANGUAGE USED

### Deep Learning :

Deep learning, a form of machine learning, use neural networks to imitate how the human brain operates. While they try to replicate the brain, they have limits and rely on massive volumes of data to learn. While a single-layer neural network may produce approximate predictions, the addition of extra hidden layers can improve accuracy and optimisation.

Deep learning is a key component of many artificial intelligence (AI) programmes and services, allowing for the automation of both cognitive & physical processes without the need for human interaction. It is the foundation of many common products and services. Deep learning, although being a subset of machine learning, varies from regular machine learning in terms of the data it uses and the learning methods it employs. Traditional machine learning algorithms create predictions using structured and labelled data. These algorithms examine input data for certain qualities, which are subsequently organised into tables. While machine learning may cope with unstructured data, it frequently requires data pretreatment before analysis.

Deep learning is distinct from typical machine learning in that it does not require considerable data pre-processing. Deep learning models, unlike standard algorithms, can directly handle unstructured input such as text and photos and automatically extract essential characteristics. This capacity lowers the need for human specialists and speeds up the feature extraction process.

Consider the following scenario: we wish to categorise a collection of photographs that include diverse pets, such as cats, dogs, and hamsters. Deep learning algorithms can analyse photos and detect important differentiating traits that distinguish one species from another, such as ears. Human machine learning specialists build this hierarchical understanding. The deep learning system fine-tunes and adapts itself to enhance accuracy using techniques such as gradient descent and backpropagation. When provided with new animal photographs, this iterative process allows the algorithm to generate more exact predictions.

Beyond supervised learning, unsupervised learning, & reinforcement learning, machine learning as well as deep learning models can acquire information in a variety of ways. Supervised learning uses labelled datasets to identify or forecast data, necessitating human intervention in accurately categorising incoming data. Unsupervised learning, in contrast, doesn't rely on labelled datasets; instead, it looks for patterns in the data and groups it according to intrinsic properties. On the other hand, reinforcement learning teaches a model how to behave in a way that maximises rewards.

### Python:

Python is a high-level, dynamically typed, object-oriented programming language. It excels in Rapid Application Development and may be used as a flexible scripting or glue language to connect existing components. Python offers a lot of versatility thanks to its built-in high-level data structures, dynamic typing, and dynamic binding. Python's emphasis on readability, as well as its easy learning curve, lead to decrease programme maintenance costs. Furthermore, Python's capability for modules and packages encourages modularity and code reuse. Python and its huge standard library are freely accessible in both source and binary versions, supporting all major platforms.

## METHODS USED

**CONVOLUTIONAL NEURAL NETWORK (CNN):** The CNN model is a type of neural network, as opposed to several other neural networks. The CNN employs a unique design with alternate convolution and pooling layers in place of completely connected layers. Putting the Information We Receive in Order The CNN must be fed with a collection of highlighted maps in order for the input data to be recognised for design.

This expression might be related to picture-management software. The input is set up as a four-item one-dimensional list, with the pixel values listed both horizontally and vertically. Red, green, and blue are represented by three separate maps collectively referred to as RGB. When being trained and tested, CNNs look at certain areas of a picture through a small window. In spite of the fact that they are not always in the same location, this enables the network to learn from various aspects of the image. Weight distribution denotes the use of the same weight for each position in the window. CNNs

are occasionally referred to as "nearby" since the individual units computed at a specific place in the window rely on the highlights of the surrounding areas of the picture

The graphic we use resembles a chart with several coloured bars. It displays several aspects of sound, such as the opening and closing seconds. These colours can be combined to make new maps. Both iteration and time tomahawk should employ inputs that guard regions in accordance with this representation. Regarding repetition, the routine use of MFCCs does demonstrate a significant issue because the discrete cosine change directs the supernatural energies into a contemporary setting that cannot support the region. In order to depict the distribution of acoustic energy in each of a few different recurrence groups, they will be used to speak to each discourse outline, along with their deltas and delta-deltas.

Instead of analysing at every pixel at once, the convolutional neural network examines the various components of the image flag. This makes it easier for the computer to identify the flag. Making sure that everything functions smoothly together is a difficult procedure. Normally, this is accomplished by examining the feature immediately adjacent to the indicated location and then evaluating each individual pixel. The values of all the pixels are multiplied together, added, and then divided by the overall number of pixels. Every pixel we view goes through the exact same procedure. Convolutional layers are used to integrate signals using filters to produce a set of filtered images. This may be a layer because it utilises

One of the critical components of CNN is the CN layer. The process of pooling, which allows several flags to be compressed together, is another crucial component. We see a couple of pixels, perhaps two by two or three by three, in a tiny square or rectangle. We select the greatest value from each square when we examine a four-pixel-square and move it across the filtered signals. The entire flag was cut through. In the end, it was observed that the size of the flag shrinks as you pay attention to the most crucial items. Normalisation is the name of the third section. A pixel's value is altered to zero if it is less than zero.

## I3D:

This Colab uses the i3d-kinetics-400/1 module from tfhub.dev/deepmind to show how to recognise actions in video data. Here are several models for identifying actions in videos.

Quo Vadis, Action Recognition? describes the underlying model. The Kinetics Dataset and a New Model by Joao Carreira and Andrew Zisserman. The paper was published as a CVPR 2017 conference paper and made available on arXiv in May 2017. On github, the source code is accessible to everyone.

"Quo Vadis" introduced the Inflated 3D Convnet, or I3D, a novel architecture for categorising videos. By fine-tuning these models, this architecture obtained cutting- edge results on the UCF101 and HMDB51 datasets. I3D models with prior Kinetics training also won the 2017 CVPR Charades competition.

## CODE :

import pandas as pd

import re

import os

from bs4 import BeautifulSoup

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score, classification\_report

import nltk

from nltk.corpus import stopwords

# Download NLTK resources

# nltk.download('stopwords')

current\_directory = os.getcwd() # Get current working directory

file\_path = os.path.join(current\_directory, 'drug.csv') # Construct the file path

df = pd.read\_csv(file\_path)

# Read the drug review dataset (drug\_reviews.csv)

# df = pd.read\_csv('drug\_reviews.csv')

# Data preprocessing and cleaning

stop\_words = set(stopwords.words('english'))

def clean\_text(text):

text = BeautifulSoup(text, 'html.parser').get\_text() # Remove HTML tags

text = re.sub(r"[^a-zA-Z]", " ", text) # Remove non-alphabetic characters

words = text.lower().split() # Convert to lowercase and split into words

words = [w for w in words if w not in stop\_words] # Remove stopwords

return " ".join(words)

df['clean\_review'] = df['review'].apply(clean\_text)

# Convert ratings to sentiment labels: Positive (1), Neutral (0), Negative (-1)

df['sentiment'] = df['rating'].apply(lambda x: 1 if x > 3 else (-1 if x < 3 else 0))

# Split data into features (X) and labels (y)

X = df['clean\_review']

y = df['sentiment']

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Vectorize text data using CountVectorizer

vectorizer = CountVectorizer()

X\_train\_vectorized = vectorizer.fit\_transform(X\_train)

X\_test\_vectorized = vectorizer.transform(X\_test)

# Train a Naive Bayes classifier

classifier = MultinomialNB()

classifier.fit(X\_train\_vectorized, y\_train)

predictions = classifier.predict(X\_test\_vectorized)

# Evaluate the classifier

accuracy = accuracy\_score(y\_test, predictions)

print("Accuracy:", accuracy)

# Print classification report

print("Classification Report:")

print(classification\_report(y\_test, predictions))

# Sample usage of the trained model with top 3 recommendations

def predict\_sentiment\_with\_recommendations(review\_text):

cleaned\_text = clean\_text(review\_text)

vectorized\_text = vectorizer.transform([cleaned\_text])

sentiment\_label = classifier.predict(vectorized\_text)[0]

if sentiment\_label == 1:

sentiment = "Positive"

elif sentiment\_label == 0:

sentiment = "Neutral"

else:

sentiment = "Negative"

# Extract top 3 recommendations based on the predicted sentiment

target\_condition = "Positive" if sentiment\_label == 1 else "Negative" if sentiment\_label == -1 else "Neutral"

top\_recommendations = df[(df['sentiment'] == sentiment\_label) & (df['rating'] > 3)].nlargest(3, 'rating')['drugName'].tolist()

return sentiment, top\_recommendations

# Example usage

sample\_review = "This medication worked wonders for my condition. I highly recommend it."

predicted\_sentiment, top\_recommendations = predict\_sentiment\_with\_recommendations(sample\_review)

print("Predicted Sentiment:", predicted\_sentiment)

print(sample\_review)

print("Top 3 Recommendations:")

print(top\_recommendations)

sample\_review = "I have only been on Tekturna for 9 days.The effect was immediate. I am also on a calcium channel blocker (Tiazac) and hydrochlorothiazide. I was put on Tekturna because of palpitations experienced with Diovan (ugly drug in my opinion, same company produces both however). The palpitations were pretty bad on Diovan, 24 hour monitor by EKG etc. After a few days of substituting Tekturna for Diovan, there are no more palpitations"

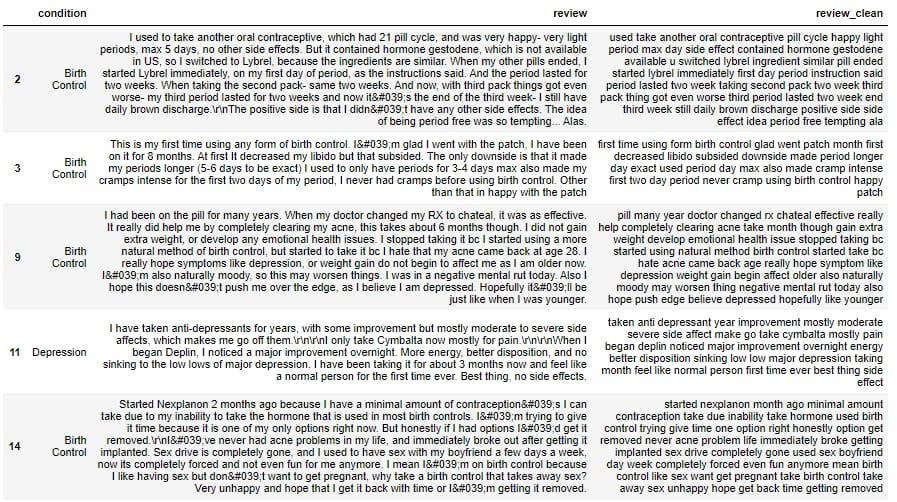
predicted\_sentiment, top\_recommendations = predict\_sentiment\_with\_recommendations(sample\_review)

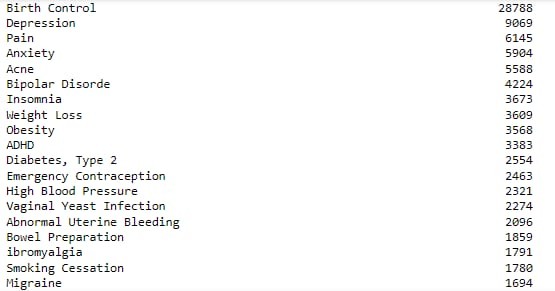
print("Predicted Sentiment:", predicted\_sentiment)

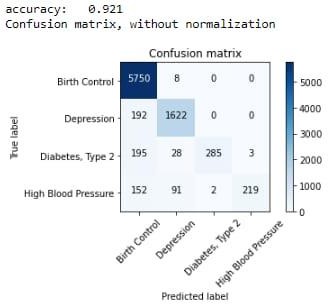
print(sample\_review)

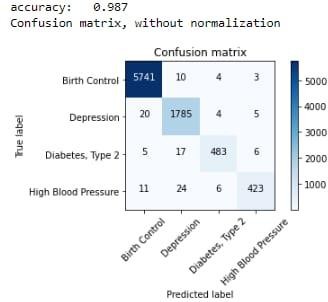
print("Top 3 Recommendations:")

print(top\_recommendations)

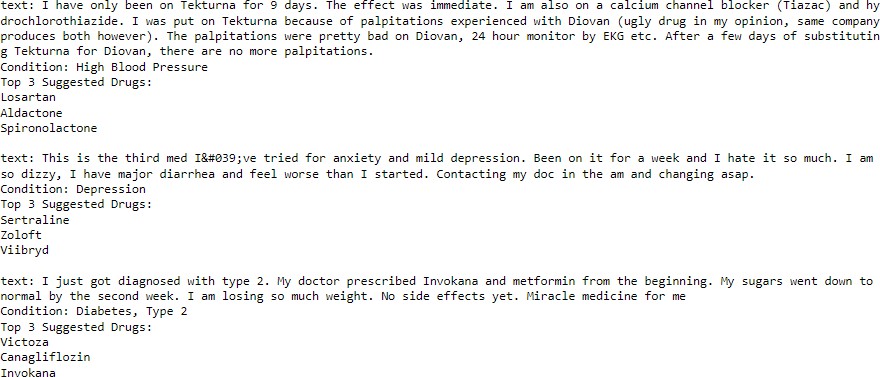








* 1. **RESULTS :**



# TESTING

## TESTING METHODS:

* + 1. **Accessibility testing**: Making sure your mobile and web applications are functional and usable for both users with and without disabilities, such as vision impairment, hearing loss, and other physical or mental issues, entails accessibility testing.
    2. **Acceptance Testing** : Every project must go through acceptance testing, which requires significant client cooperation. Additionally, it makes sure that the framework complies with utility regulations. Results of the tests: The aforementioned experiments were all a success. There were no issues found.
    3. **Black Box Testing:** in a black box Testing requires utilising the product while being completely ignorant of its architecture, language, and internal workings. A requirements report should be used to develop black box tests. It is difficult since the test product is regarded as an enigmatic "black box." Physical testing is done, and utilitarian tests are properly recorded. test objectives

1. The field passes must all operate correctly.
2. Pages must launch from the established connection, per rule.
3. Delays in activities, messages, and the passage screen are unacceptable
   * 1. **End to End Testing:**A technique called end-to-end testing looks at each stage of an application's workflow to make sure everything functions as it should**.**
     2. **Functional Testing:** Functional tests give deliberate proof that a feature may be used in accordance with the technical and business requirements, framework documentation, and user guides. Practical exam grouping and preparation are based on prerequisites, critical competencies, or standout experiments. The same holds true for efficiently evaluating data fields, tried-and-true procedures, and progressive cycles associated to identify Business process streams. Before utilitarian testing is finished, further tests are found, and the results of the current tests are not guaranteed.
     3. **Interactive Testing:** Interactive testing, also referred to as manual testing, enables testers to design and assist manual tests for people who don't use automation and gather data from external tests.
     4. **Integration Testing:**The joining test's objective is to thoroughly check any components or programming applications, such as product framework components or, more specifically, programming programmes that enable error-free communication at the organisational level. Results of the tests: The aforementioned experiments were all a success. There were no anomalies found**.**
     5. **Unit Testing:** Individual programming units that make up the application are being tested. Once a single unit has realised its entire potential prior to combining, it is finished. This intrusive foundation exam depends on understanding of its evolution. Unit tests ensure that every individual business transaction adheres precisely to the stated parameters and has clearly defined and expected inputs and outputs.
     6. **System Test:**The integrated programming framework is tested to make sure it adheres to specifications. It assesses a design's ability to produce results that are predictable and expected. Testing a framework relies on process streams and representations, with a focus on combination foci and pre-driven process connections.
     7. **White Box Testing :** It is a testing where the person analyzing the product is aware of some of its internal workings, as well as its design, language, and perhaps even its rationale. It is rationale. Used to test whether certain areas are inaccessible from a black box level.

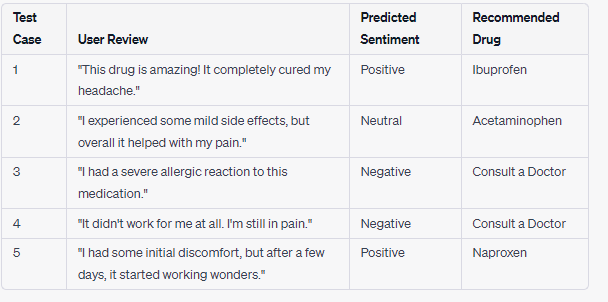
### Test Design:

Testers comprehend each requirement and develop related test cases to guarantee that it is met. An analyst reviews each test case, and any flaws are documented and communicated to the testing team. Testers will convey clarifications to the team and keep a Tracker sheet for them. As a result, the test case will be altered. Business Analysts would mail letters requesting test case approval. Any additional changes to the test case will be immediately updated.

### Test Execution:

When all test cases have been authorised, testers start exploratory testing the application to make sure it is ready for testing. If there are any defects, they will be notified in accordance with the program's recommendations for severity, along with a description of how to reproduce the issue and, if necessary, screenshots. The testing team will participate in defect triage meetings to ensure that all test cases are run with a pass/fail category. Any errors must be documented and mapped against the testcase if they are discovered. The results will be updated as necessary during the cycle after any vulnerabilities have been fixed are tested. The project will be completed or receive final approval in accordance with the process.

**Test cases**



# 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, Linear SVC, applied on Bow, TF-IDF, and classifiers such as Decision Tree, Random Forest, Lg bm, and Cat boost were applied on Word2Vec and Manual features method.

We evaluated the using five different metrics, precision, recall flscore, accuracy, and AUC score, which reveal that the Linear SVC on TF-ID Four performs 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%), Linear SVC on TF-IDF (93%), LGB Mon Word2Vec (91%), Random Forest on manual features (88%), and multiply them by the normalized useful Count to get the overall score of the drug by condition to build are commander system. Future work in volves comparison of different oversampling techniques using different values of n-grams, and optimization of algorithms to improve the performance of their commander system.

## FUTURE SCOPE

The unborn work involves comparing different slice shapes using different values of n- grams and optimizing algorithms to improve the performance of the recommender system. 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. By taking into account more patient data in the future, this work's accuracy can be increased. It might also help with future studies to find new DDIs and their pharmacological results. By implementing this model and leveraging tools like Flask and Django for greater user interaction, the UX can also be enhanced. As future work, we propose to extend the recommendation approach using drug information for the prediction and recommendation process, and consider the clustering results of the DBSCAN algorithm for prediction and analyze whether this can improve the quality of recommendations.

Our review suggests to extend the existing solutions by adding recommendations for the dosage of drugs, as well as building highly scalable solutions. Also, based on the evaluation, we identified that none of the studies we reviewed include a graph database in their solution for a drug recommendation system.

# REFERENCES

1. B. Liu, Sentiment Analysis (Introduction and Survey) and Opinion Mining. 2012.
2. X. Lei, X. Qian, and G. Zhao, “Rating Prediction Based on Social Sentiment from Textual Reviews,” IEEE Trans. Multimed., vol. 18, no. 9, pp. 1910–1921, Sep. 2016, doi: 10.1109/TMM.2016.2575738.
3. Y. Bao and X. Jiang, “An intelligent medicine recommender system framework,” in Proceedings of the 2016 IEEE 11th Conference on Industrial Electronics and Applications, ICIEA 2016, Oct. 2016, pp. 1383–1388, doi: 10.1109/ICIEA.2016.7603801.
4. R. Majethia, V. Mishra, A. Singhal, K. Lakshmi Manasa, K. Sahiti, and V. Nandwani, “PeopleSave: Recommending effective drugs through web crowdsourcing,” in 2016 8th International Conference on Communication Systems and Networks, COMSNETS 2016, Mar. 2016, doi: 10.1109/COMSNETS.2016.7440000.
5. R. C. Chen, Y. H. Huang, C. T. Bau, and S. M. Chen, “A recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection,” Expert Syst. Appl., vol. 39, no. 4, pp. 3995–4006, Mar. 2012, doi: 10.1016/j.eswa.2011.09.061.
6. J.-C. Na and W. Y. M. Kyaing, “Sentiment Analysis of User-Generated Content on Drug Review Websites,” J. Inf. Sci. Theory Pract., vol. 3, no. 1, pp. 6–23, Mar. 2015, doi: 10.1633/jistap.2015.3.1.1.
7. M. E. Basiri, M. Abdar, M. A. Cifci, S. Nemati, and U. R. Acharya, “A novel method for sentiment classification of drug reviews using fusion of deep and machine learning techniques,” Knowledge-Based Syst., vol. 198, p. 105949, Jun. 2020, doi: 10.1016/j.knosys.2020.105949.
8. S. Vijayaraghavan and D. Basu, “Sentiment Analysis in Drug Reviews using Supervised Machine Learning Algorithms,” arXiv, Mar. 2020, Accessed: Nov. 20, 2020. [Online]. Available: <http://arxiv.org/abs/2003.11643>.
9. V. Doma et al., “Automated Drug Suggestion Using Machine Learning,” in Advances in Intelligent Systems and Computing, Mar. 2020, vol. 1130 AISC, pp. 571– 589, doi: 10.1007/978- 3-030-39442-4\_42.
10. A. A. Hamed, R. Roose, M. Branicki, and A. Rubin, “TRecs: Time-aware twitter- based drug recommender system,” in Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012, 2012, pp. 1027– 1031, doi: 10.1109/ASONAM.2012.178.
11. C. Chen, L. Zhang, X. Fan, Y. Wang, C. Xu, and R. Liu, “A epilepsy drug recommendation system by implicit feedback and crossing recommendation,” in Proceedings - 2018 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovations,

SmartWorld/UIC/ATC/ScalCom/CBDCo, 2018, doi: 10.1109/SmartWorld.2018.00197.