A Project Report on

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

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

A Shashi kumar

(20H51A0555)

R Grishma

(20H51A0573)

S Tharun kumar

(20H51A0576)

Under the esteemed guidance of

B. SIVAIAH

(Associate Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report entitled **" Drug Recommendation System On Sentiment Analysis Of Drug Reviews Using Machine Learning "** being submitted by A. Shashi Kumar (20H51A0555), Rajuru Grishma (20H51A0573) and S.Tharun (20H51A0576) in partial fulfillment for the award of**Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. B. Sivaiah
Associate Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Mr. B. Sivaiah,** Associate Professor, Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**, Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Dept Name for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and Shri Ch Abhinav Reddy, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

A. SHASHI – 20H51A0555

R. GRISHMA– 20H51A0573

S. THARUN – 20H51A0576

TABLE OF CONTENTS

List of Figures

ABSTRACT

In today's digital age, healthcare is a crucial aspect of the medical field. With the internet, people have access to a vast amount of health-related information, but it's important to be cautious of misinformation. Unfortunately, the sheer volume of clinical information scattered across various websites can make it challenging for users to find reliable information to improve their well-being. Medication errors pose a significant threat to patients' lives, underscoring the need for recommendation systems in healthcare to help users make informed decisions. We focus on developing drug recommendation systems to assist users in identifying the right medications for specific health conditions.

These systems analyze reviews from other users on different medications for various diseases. Our goal is to utilize data mining concepts, visualization, and sentiment analysis to recommend drugs based on factors like the user's condition, ratings, and reviews. We employ machine learning approaches, as well as content and collaborative filtering methods, to tailor recommendations for each patient's health condition. In simpler terms, imagine you're searching online for information about a health issue you're dealing with. You find tons of websites with lots ofinformation, but it's overwhelming and hard to know what's trustworthy. Plus, when it comes to medications, you want to make sure you're getting the right ones for your specific condition. That's where our drug recommendation system comes in. It sifts through all the information out there, looks at what other people have said about different medications, and uses advanced techniques to suggest the best options for you based on your condition and what's worked well for others in similar situations. We'll explain how we gather and analyze this data, visualize it to make it easier to understand, and use it to recommend medications that are most likely to help you. By using these methods, we aim to make it simpler for people to make informed decisions about their health and avoid potentially harmful mistakes when it comes to medication.

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1. Problem Statement

In the modern healthcare landscape, the abundance of health-related information available on the internet has created a double-edged sword for users. While the accessibility of this information holds immense potential for empowering individuals to make informed decisions about their health, the sheer volume and lack of quality control pose significant challenges. Users often struggle to navigate through the plethora of online resources to find accurate and trustworthy information relevant to their specific health concerns. This issue is further compounded when it comes to medication, as errors in selecting the right treatment can have serious consequences, including threats to patients' lives. Moreover, the traditional approach of relying solely on healthcare professionals for guidance is often insufficient in addressing the diverse and evolving healthcare needs of individuals. Patients increasingly seek to complement professional advice with insights from peer experiences and recommendations. Therefore, there is a critical need to develop effective recommendation systems tailored to the healthcare domain. These systems must leverage advanced technologies such as data mining, sentiment analysis, and machine learning to sift through vast amounts of health-related data, including user reviews of medications.By doing so, these systems can assist users in making more informed, efficient, and accurate health-related decisions, particularly regarding medication selection for specific health conditions. However, designing and implementing such recommendation systems present several challenges. These include but are not limited to:

Data Quality and Trustworthiness: Ensuring the reliability and accuracy of the data collected from various online sources, including user-generated reviews, is essential to avoid propagating misinformation and potential harm.

Personalization and Contextualization: Developing algorithms capable of understanding the unique health profiles and preferences of individual users to provide personalized recommendations tailored to their specific needs and circumstances.

User Interface and Accessibility: Designing user-friendly interfaces that simplify the navigation of health-related information and facilitate seamless interaction with the recommendation system, particularly for users with limited technical expertise.

Ethical and Legal Considerations: Addressing privacy concerns and complying with regulations regarding the collection, storage, and usage of sensitive health data while maintaining transparency and accountability in the recommendation process.

In light of these challenges, the primary objective of this research is to develop and evaluate a robust drug recommendation system that leverages cutting-edge technologies to empower users in making informed medication choices based on reliable data and peer insights. By addressing these challenges, the proposed recommendation system aims to enhance the accessibility, efficiency, and safety of healthcare decision-making in the digital era.

1.2. Research Objective

The primary objective of this research is to develop an advanced drug recommendation system tailored to the specific needs of users navigating the vast landscape of onlinehealthcare information. This system aims to mitigate the challenges inherent in medication selection, particularly concerning the reliability and relevance of available data. Through comprehensive data collection and preprocessing efforts, diverse datasets comprising user- generated reviews and ratings of medications will be compiled, ensuring data quality and relevance. Leveraging cutting-edge algorithms grounded in data mining, sentiment analysis, and machine learning, the system will extract actionable insights from the dataset to facilitate personalized medication recommendations.

This personalization will consider individual user profiles, incorporating factors such as health history, preferences, and demographics to enhance recommendation accuracy and effectiveness. User-centric design principles will guide the creation of an intuitive and accessible interface, catering to users with varying levels of technical proficiency. Rigorous evaluation and validation processes will assess the system's performance in terms of accuracy, relevance, and user satisfaction, ensuring alignment with user needs and expectations. Ultimately, this research aims to empower users in making informed medication choices, thereby contributing to improved healthcare decision-making and patient outcomes in the digital age.

1.3. Project Scope and Limitations

Scope:

The scope of this project encompasses the design, development, and implementation of a drug recommendation system tailored to the healthcare domain. Key components and activities within theproject scope include:

Data Collection and Preprocessing: Gathering a diverse dataset comprising user- generated reviews and ratings of medications from various online sources. This includes ensuring data quality, relevance, and compliance with ethical standards and legal regulations.

Algorithm Development: Designing and implementing advanced algorithms leveraging data mining, sentiment analysis, and machine learning techniques to analyze the dataset and generate medication recommendations. This involves developing algorithms for personalized recommendation based on individual user profiles and contextual factors.

User Interface Design: Designing an intuitive and user-friendly interface for the recommendation system. This includes wireframing, prototyping, and iterative design to optimize usability and accessibility for users with varying levels of technical proficiency.

System Implementation: Developing the recommendation system software based on the designed algorithms and user interface specifications. This includes coding, testing, and integration of system components to ensure functionality and reliability.

Evaluation and Validation: Conducting rigorous evaluation and validation of the recommendation system to assess its performance in terms of accuracy, relevance, and user satisfaction. This involves comparing the system's recommendations against expert judgments, conducting user studies, and soliciting feedback for iterative improvement.

Ethical and Legal Compliance: Ensuring compliance with ethical standards and legal regulations governing the collection, storage, and usage of sensitive health data. This includes implementing measures to safeguard user privacy and security throughout the recommendation process.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 A Machine learning based drug recommendation system for health care

2.1.1. Introduction

One of the most widely discussed topics on the internet is health-related information, reflecting people's increasing concern about their well-being. Studies, such as the Pew Internet survey, reveal that 55% of internet users seek health-related information online. Researchers have also analyzed search terms entered search engines to understand common health-related queries. Additionally, according to a report from NCBI, around 99,000 people die annually due to mistakes made by medical professionals in hospitals. These alarming statistics highlight the urgent need for recommendation systems in healthcare to help users make better-informed decisions about their health. Recommendation systems, leveraging technologies like Machine Learning and Data Mining, are crucial in today's rapidly advancing technological landscape, as they can potentially save lives. This paper introduces a proposed drug recommendation system that utilizes patient reviews and ratings to suggest medications tailored to specific conditions. The system employs advanced technologies such as Machine Learning and Data Mining, along with a Content and Collaborative filtering approach.

Its primary objective is to design an effective and accurate system for recommending drugs to patients. With the abundance of data available on the internet, our system aims to analyze this data accurately, efficiently, and at scale to fulfill its objective.

2.1.2. Merits, Demerits and Challenges

Merits:

Improved Access to Information: The proposed drug recommendation system enhances access to health-related information, allowing users to make informed decisions about medication choices based on patient reviews and ratings.

Personalized Recommendations: By leveraging technologies like Machine Learning and Data Mining, the system provides personalized medication recommendations tailored to individual health conditions and preferences.

Potential to Save Lives: Recommendation systems in healthcare have the potential to save lives by reducing medication errors and improving the accuracy of treatment decisions.

Efficiency and Scalability: The use of advanced technologies enables the system to analyze large datasets efficiently, making it scalable to accommodate the vast amount of health-related information available online.

Demerits:

Data Quality and Reliability: The accuracy and reliability of patient reviews and ratings may vary, potentially leading to incorrect recommendations or biases in the system.

Privacy Concerns: Collecting and analyzing personal health data raises privacy concerns, necessitating stringent measures to protect user confidentiality and comply with data protection regulations.

Dependency on Technology: The effectiveness of the recommendation system relies heavily on the performance and accuracy of the underlying technologies such as Machine Learning and Data Mining. Any inaccuracies or limitations in these technologies could impact the reliability of the recommendations.

Challenges:

Data Heterogeneity: Healthcare data is often fragmented and heterogeneous, originating from various sources with varying formats and standards. Integrating and harmonizing this data presents a significant challenge.

Ethical and Legal Considerations: Adhering to ethical principles and legal regulations, such as patient privacy laws (e.g., HIPAA), while collecting and analyzing health data is paramount. Ensuring compliance with these regulations adds complexity to system development and deployment.

Bias and Fairness: Recommendation systems may inadvertently introduce biases, leading to unequal treatment or unfair outcomes for certain user groups. Addressing and mitigating bias in thesystem poses a challenge.

User Adoption and Trust: Convincing users to trust and adopt the recommendation system requires transparent communication about how recommendations are generated and ensuring that users understand the limitations and risks associated with the system. Building trust in the system is crucial for its success.

2.1.3. Implementation

The implementation of the proposed drug recommendation system involves several key steps, each utilizing various Python libraries and techniques:

Data Cleaning and Preparation: Utilize libraries such as Pandas and NumPy for data cleaning, preparation, and visualization. Perform stemming using NLTK Library to extract the base form of words and remove affixes. Implement Count Vectorizer to convert words into vector form and remove stop words from strings.

Exploratory Data Analysis (EDA): Analyze and visualize the Drug Review Dataset using graphs and charts to understand various factors and trends. Convert non-numeric columns like 'Rating' and 'Useful Count' into numeric data types for analysis. Generate metadata of drugs by concatenating drug names, conditions, and reviews.

Splitting and Processing Data: Split the Drug Review dataset into training and testing files. Remove irrelevant information from the dataset. Implement stemming using NLTK Library to extract the base form of words and ensure consistency in meaning. Utilize Count Vectorizer to convert words into vector form, considering the 'tags' column.

Logistic Regression: Apply logistic regression to understand the accuracy and validity of the dataset. Create a classifier object and fit the model on the training dataset. Perform prediction on the test dataset and evaluate the model using confusion matrix and evaluation metrics like accuracy, precision, F1 score, and recall.

Model Evaluation with Similarity and Distance Metrics: Evaluate the model using various similarity and distance metrics such as Pearson Correlation, Spearman Correlation, Cosine Similarity, Jaccard's Similarity, Euclidean Distance, and Manhattan Distance. Compare similarities between the first drug and random drugs to assess model performance.

Building Recommendation System: *Implement two approaches for recommendation:* Content-based filtering and Collaborative-based filtering. For content-based filtering, use cosine similarity to recommend drugs based on the selected condition. For collaborative-based filtering, compare user-item interactions to recommend drugs based on previous user reviews. Test the recommendation system by indicating suitable medicines for input conditions and analyzing correlations with similar conditions.

3

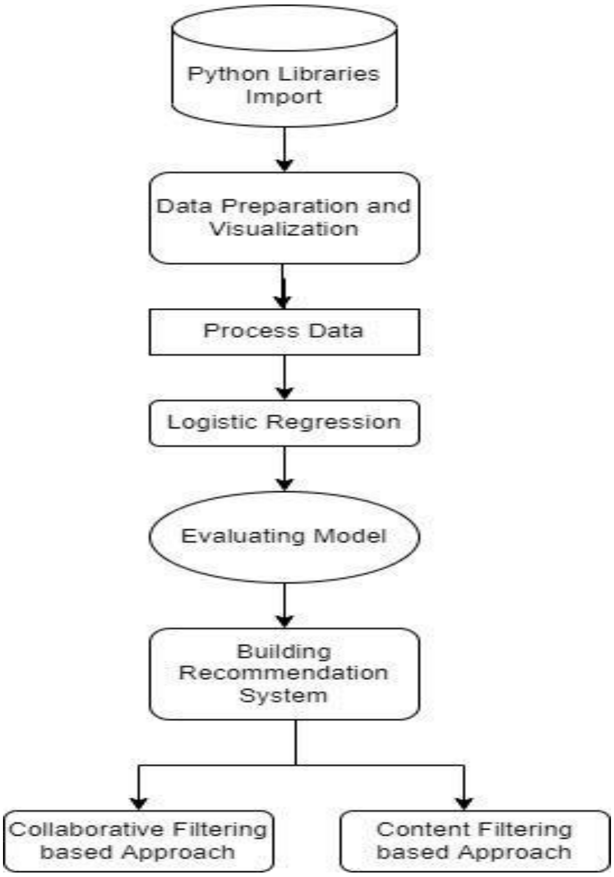


Figure 2.1: The architecture of existing system.

2.2. A Computer-based disease prediction and medicine recommendation systemusing machine learning approach

2.2.1. Introduction

The paper introduces a novel approach to disease prediction and medication recommendation using machine learning algorithms: Decision Tree Classifier, Random Forest Classifier, and Naive Bayes Classifier. Accurate disease prediction is crucial for effective treatment, and the system aims to predict diseases based on patient-reported symptoms, subsequently recommending tailored medications. The study emphasizes the importance of accurate prediction in healthcare, highlighting the detrimental effects of erroneous predictions. By combining insights from multiple classifiers, the system enhances prediction accuracy and reliability, ultimately improving patient outcomes. Overall, the paper underscores the significance of accurate disease prediction and medication recommendation in modern healthcare and demonstrates the efficacy of the proposed system through rigorous experimentation.

2.2.2. Merits, Demerits and Challenges

Merits:

Accurate Disease Prediction: The use of machine learning algorithms enables accurate prediction of diseasesbased on symptoms, enhancing the potential for correct treatment.

Ensemble Learning Approach: Combining predictions from multiple classifiers (Decision Tree, RandomForest, Naive Bayes) improves prediction accuracy and reduces the risk of misdiagnosis.

Medicine Recommendation: The system not only predicts diseases but also recommends appropriate medications based on the predicted disease, enhancing the overall treatment process.

Utilization of Advanced Techniques: The paper utilizes advanced machine learning techniques such as Decision Trees, Random Forests, and Naive Bayes classifiers, showcasing the application of cutting-edgetechnologies in healthcare.

Demerits:

Dependency on Data Quality: The accuracy of disease prediction heavily relies on the quality and completeness of the input data, which may vary and affect the reliability of predictions.

Limited Scope: The system's effectiveness may be limited to the diseases and symptoms included inthe dataset, potentially missing out on rare or emerging diseases not covered in the training data.

Interpretability: Machine learning models like Decision Trees and Random Forests may lack

interpretability, making it challenging to understand the underlying reasoning behind predictions.

Model Overfitting: Overfitting can occur, especially with Decision Trees, leading to overly complexmodels that perform well on the training data but generalize poorly to new data.

Challenges:

Data Availability: Obtaining comprehensive and high-quality medical datasets for training machinelearning models poses a challenge due to privacy concerns and data accessibility issues.

Model Evaluation: Evaluating the performance of machine learning models for disease prediction

requires robust validation methods to ensure reliability and generalizability.

Interpretability vs. Accuracy Trade-off: Balancing the need for accurate predictions with the interpretability of models is challenging, as complex models may sacrifice interpretability for improvedaccuracy.

Ethical Considerations: Ensuring ethical use of patient data and preventing biases in predictions are

critical challenges in implementing healthcare-related machine learning systems.

2.2.3. Implementation

The implementation involves training separate models using the three classifiers (Decision Tree, Random Forest, Naive Bayes) on a dataset containing symptoms and corresponding diseases. Predictions from each model are combined based on predefined rules to determine the final disease prediction. The system also recommends medications based on the predicted disease. Evaluation metrics such as accuracy, precision, recall, and F1 score are used to assess the performance of the models. Additionally, the paper discusses the formulation parameters and example calculations for the Naive Bayes Classifier to predict diseases based onsymptoms.

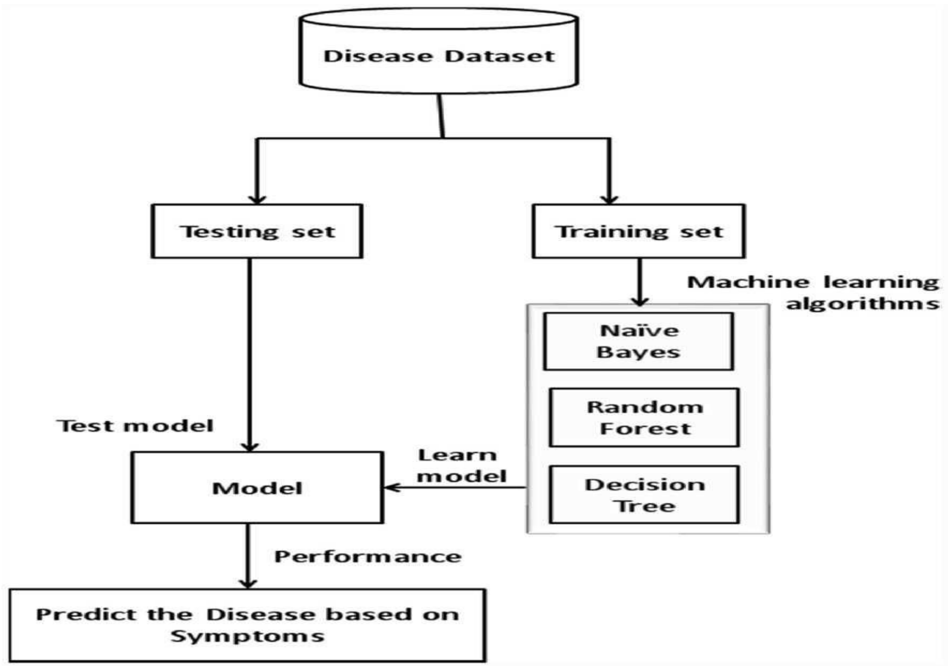


Figure 2.2 drug recommendation using ml

2.3. Drug Recommendation System for Diabetes Using a Collaborative Filtering and Clustering Approach

2.3.1. Introduction

The introduction highlights the critical need for accurate disease prediction and medication recommendation in healthcare, particularly in the context of increasing patient data complexity. Leveraging machine learning algorithms like Decision Tree, Random Forest, and Naive Bayes classifiers, the study aims to develop predictive models for disease prognosis based on patient symptoms. The dataset utilized, sourced from the UCI Machine Learning Repository, contains comprehensive patient records spanning a decade, primarily focusing on diabetes-related diseases. By employing ensemble learning techniques and robust model evaluation metrics, the study endeavors to enhance prediction accuracy and facilitate personalized treatment strategies. Ultimately, the introduction sets the stage for exploring how machine learning can revolutionize disease prediction and medication recommendation, ultimately improving patient outcomes in healthcare delivery.

2.3.2. Merits, Demerits and Challenges

Merits:

Comprehensive Dataset: The dataset provides a detailed overview of patient health records, enabling in-depth analysis and exploration of patterns and trends.

Exploratory Data Analysis (EDA): EDA helps in understanding the dataset, identifying patterns, anomalies, and testing hypotheses, thereby laying a strong foundation for subsequent analysis.

***Data Transformation*:** Converting categorical variables to binary variables and creating new variables from existing categories enhances the dataset's usability and analysis potential.

Variable Selection: Discarding noninformative features and selecting relevant variables streamlines the dataset, improving the efficiency and accuracy of subsequent analyses.

Demerits:

Data Cleaning Challenges: Removing duplicated cases and records with missing values can be time-consuming and may result in information loss if not executed meticulously.

Limited Medication Information: Eliminating drugs with minimal usage or those not administered to any patient may reduce the dataset's completeness and overlook potential insights from less common medications.

Challenges:

Handling Missing Data: Dealing with missing values and ensuring data integrity during thecleaning process is crucial for accurate analysis.

Feature Relevance: Determining the relevance of features and selecting appropriate variables foranalysis requires careful consideration to avoid bias and ensure meaningful insights.

Dataset Size: Managing large datasets with extensive records and variables poses computationalchallenges and may necessitate efficient processing and storage solutions.

2.3.3. Implementation

The implementation involves data preprocessing steps such as data cleaning, transformation, and variable selection to prepare the dataset for subsequent analysis. Exploratory Data Analysis (EDA) techniques are employed to understand the dataset's characteristics and identify relevant patterns. Additionally, data transformation techniques are applied to convert categorical variables, create new variables, and streamline the dataset. Variable selection ensures the inclusion of informative features while discarding irrelevant or redundant ones. Overall, the implementation focuses on optimizing the dataset for accurate disease prediction and medication recommendation based on patient records.

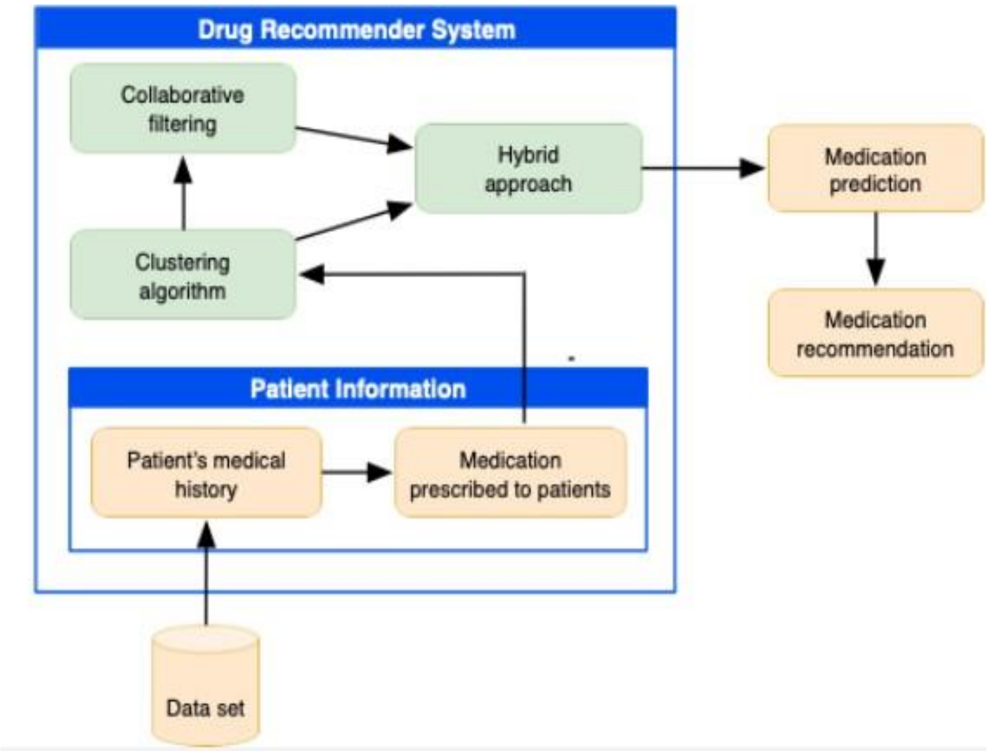


Figure 2.2 drug recommendation using filtering

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

3.1. Objective of Proposed Model:

The main objective of the proposed model for Drug Recommendation System On Sentiment Analysis Of Drug Reviews Using Machine Learning is to develop an advanced system for predicting medicines based on a patient's condition description. This system will leverage natural language processing (NLP) techniques to analyze and interpret the textual descriptions provided by patients regarding their health conditions. By understanding the nuances of these descriptions, the system will recommend suitable medicines that align with the patient's needs and health status.

The primary goal of this system is to assist healthcare professionals, including doctors, nurses, and pharmacists, in making informed decisions regarding the selection of medicines for their patients. By providing accurate and personalized medicine recommendations, the system aims to improve the overall quality of patient care and treatment outcomes.

The system will utilize state-of-the-art NLP algorithms and machine learning models, such as TF-IDF (Term Frequency-Inverse Document Frequency) and SVM (Support Vector Machine), to analyze patient descriptions and predict suitable medicines. It will also incorporate a user-friendly interface that allows healthcare professionals to easily input patient descriptions and receive accurate and timely medicine recommendations.

Overall, the development of this system represents a significant advancement in healthcare technology, offering a practical and efficient solution for medicine prediction based on patient condition descriptions. Its implementation has the potential to revolutionize the way healthcare professionals approach medication selection, ultimately leading to improved patient care and outcomes.

3.2. Algorithms Used for Proposed Model:

The proposed model utilizes TF-IDF, Support vector machine and random forest

3.2.1. TF-IDF: TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents, often used in text mining and information retrieval.

Term Frequency (TF): It measures how frequently a term appears in a document. It is calculated as the number of times a term appears in a document divided by the total number of terms in the document.

Inverse Document Frequency (IDF): It measures how important a term is by checking how often it appears across multiple documents. It is calculated as the logarithm of the total number of documents divided by the number of documents containing the term.

TF-IDF Calculation: The TF-IDF score for a term in a document is calculated by multiplying the term's TF value by its IDF value. This score helps to identify the most relevant terms in a document.

Usage in Text Classification: TF-IDF is commonly used in text classification tasks to convert text data into numerical form. Each document is represented as a vector of TF-IDF values for each term in the vocabulary. This representation allows machine learning models to work with text data effectively.

Normalization: Sometimes, the TF-IDF values are normalized to have a unit length vector. This is done to ensure that longer documents do not have an advantage due to their length.

TF-IDF is a powerful tool for extracting important features from text data and is widely used in various natural language processing applications such as document classification, information retrieval, and text summarization.

- **SVM**

Support Vector Machines (SVMs) are powerful supervised learning models used for classification and regression tasks. They work by finding the hyperplane that best separates different classes in the feature space.

How SVM Works:

Linear Separability: SVM is based on the idea of finding a hyperplane that best separates the data points of different classes. If the data is not linearly separable, SVM uses a kernel trick to map the data into a higher-dimensional space where it is linearly separable.

Maximizing Margin: The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class (support vectors).

Soft Margin: In cases where the data is not perfectly separable, SVM uses a soft margin approach, allowing some data points to be on the wrong side of the margin or even on the wrong side of the hyperplane. The goal is to find the balance between maximizing the margin and minimizing the classification error.

Kernel Trick: SVM can efficiently handle non-linearly separable data by using a kernel function to map the data into a higher-dimensional space where it is linearly separable. Common kernels include the linear, polynomial, and radial basis function (RBF) kernels.

❖ **RANDOM FOREST:**

Random Forest is a versatile and widely used ensemble learning algorithm in machine learning. It is known for its robustness, flexibility, and effectiveness across various types of datasets and predictive tasks.

Applications of Random Forest: Random Forest has wide-ranging applications across various domains, including:

- **Classification**: Random Forest is commonly used for classification tasks, such as spam detection, medical diagnosis, and customer churn prediction. Its ability to handle high-dimensional data and complex decision boundaries makes it well-suited for a wide range of classification problems.
- **Regression**: Random Forest can also be applied to regression tasks, such as stock price prediction, housing price estimation, and demand forecasting. Its robustness to outliers and nonlinear relationships in the data makes it an effective tool for predicting continuous variables.
- **Feature Importance**: By analyzing the relative importance of features based on their

contribution to the performance of the model, practitioners can gain insights into the underlying factors driving the prediction.

Random Forest is a powerful ensemble learning algorithm that combines the strengths of decision trees with the benefits of ensemble learning. Its ability to handle complex datasets, prevent overfitting, and provide insights into feature importance makes it a valuable tool in predictive modeling and data analysis.

3.3. Designing:

3.3.1. Architecture:

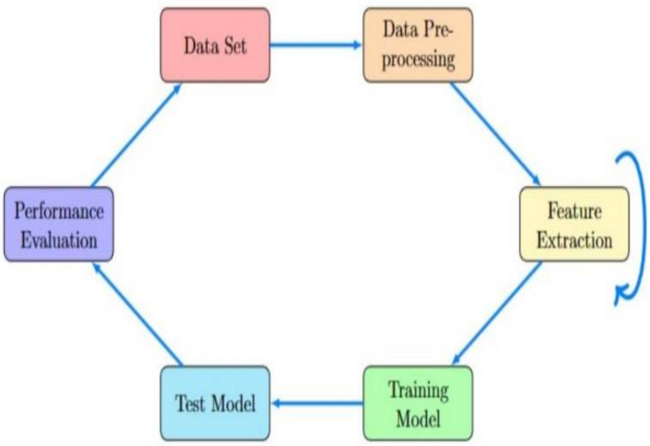


Figure.3.3.1: Architecture of the Proposed System

3.3.2. Sequence Diagram:

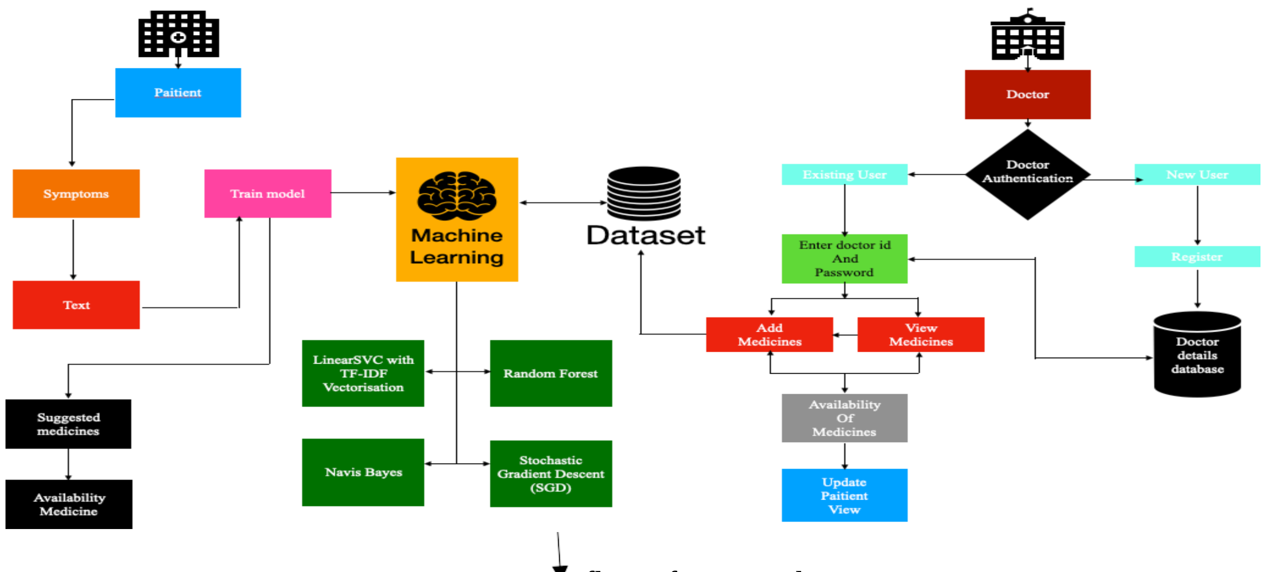


Figure.3.3.2: Sequence flow of Proposed System

3.4. Stepwise Implementation:

In this work, we will build a machine learning module. The model works on the concept of TF-IDF (Term Frequency-Inverse Document Frequency) and machine learning algorithms including SVM (Support Vector Machine), Random Forest, and Naive Bayes.

Data Preprocessing:

Before training the models, we preprocess the patient condition text data by removing stopwords, punctuation, and converting the text to lowercase. We then tokenize the text and apply TF-IDF vectorization to convert the text into numerical features.

Model Training and Evaluation:

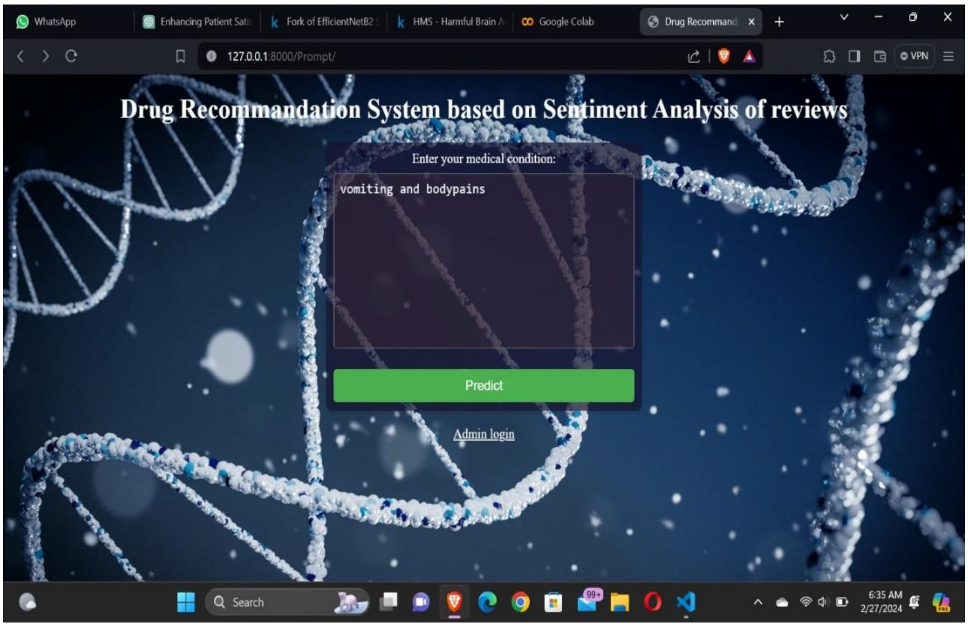
We train three different machine learning models on the preprocessed data: SVM, Random Forest, and Naive Bayes. We use the TF-IDF vectorized features as input to the models. The models are evaluated using cross-validation to ensure robustness and generalization.

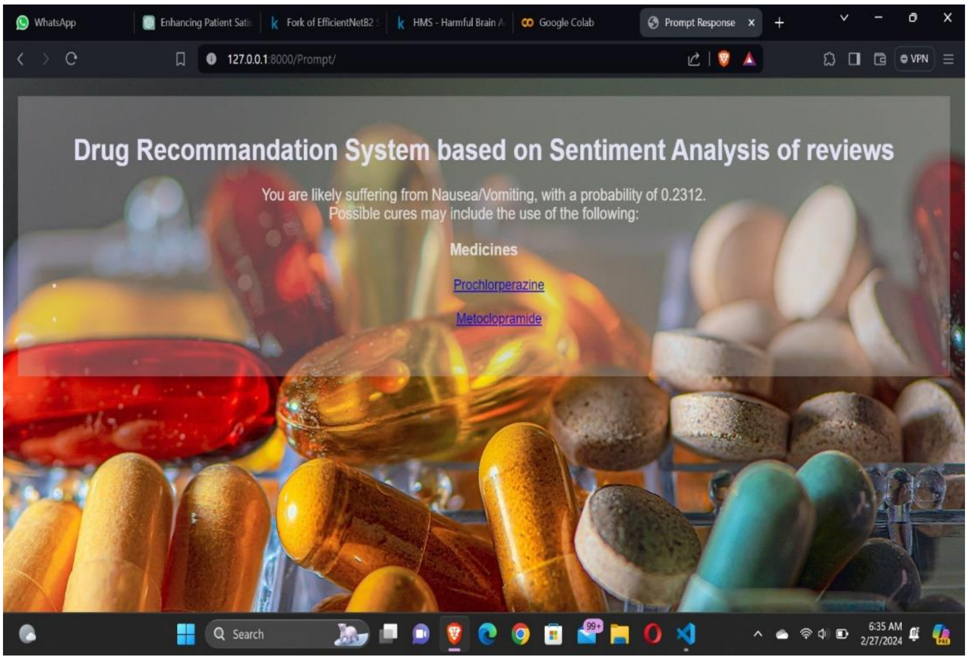
3.5 Implementation Code and output:

display.py:

```
from joblib import load
import numpy as np
from django.http import HttpResponse
from django.shortcuts import render

label_encoder  = load('Prompt/ML/label_encoder')
svc            = load('Prompt/ML/nb')
vectorizer     = load('Prompt/ML/vectorizer')
sgd            = load('Prompt/ML/sgd')
medications    = load('Prompt/ML/medications')
def predict(x, request):
    x = vectorizer.transform([x])
    predictions = sgd.predict_proba(x) + svc.predict_proba(x)
    probability = np.max(predictions)/2
    disease = label_encoder.inverse_transform([np.argmax(predictions)])
    if probability<0.1:
```

```
        return HttpResponse('Not enough information')
    context = {
        'Disease': disease[0],
        'probability': probability,
        'my_array': set(medications[disease[0]]),
    }
    return render(request, 'Prompt/prompt_response.html',context)
```





manage.py:

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'majorproject.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

Views.py:

```
from django.shortcuts import redirect
from django.urls import reverse
from django.shortcuts import render, redirect
from .forms import MyForm, LoginForm
from .ML.display import predict
from joblib import load, dump
from django.http import HttpResponse
```

```
import os
from django.contrib import messages
import re
from .forms import CreateAccountForm


description = load('Prompt/ML/description')

file_path = 'Prompt/ML/login_cred.joblib'
user_details_path = 'Prompt/ML/user_details.joblib'
medicine_set_path = 'Prompt/ML/medicine_set.joblib'
medicine_store_path = 'Prompt/ML/medicine_store.joblib'

if os.path.exists(file_path) and os.path.exists(medicine_store_path) and os.path.exists(medicine_set_path) and os.path.exists(user_details_path):
    login_cred = load(file_path)
    user_details = load(user_details_path)
    medicine_store = load(medicine_store_path)
    medicine_set = load(medicine_set_path)

else:
    login_cred = {}
    user_details = {}
    medications = load('Prompt/ML/medications')
    medicine_set = {medicine for medication in medications.keys() for medicine in medications[medication]}
    medicine_store = {}

    dump(login_cred, file_path)
    dump(medicine_set, medicine_set_path)
    dump(user_details, user_details_path)
    dump(medicine_store, medicine_store_path)
    dump(medicine_set,medicine_set_path)
```

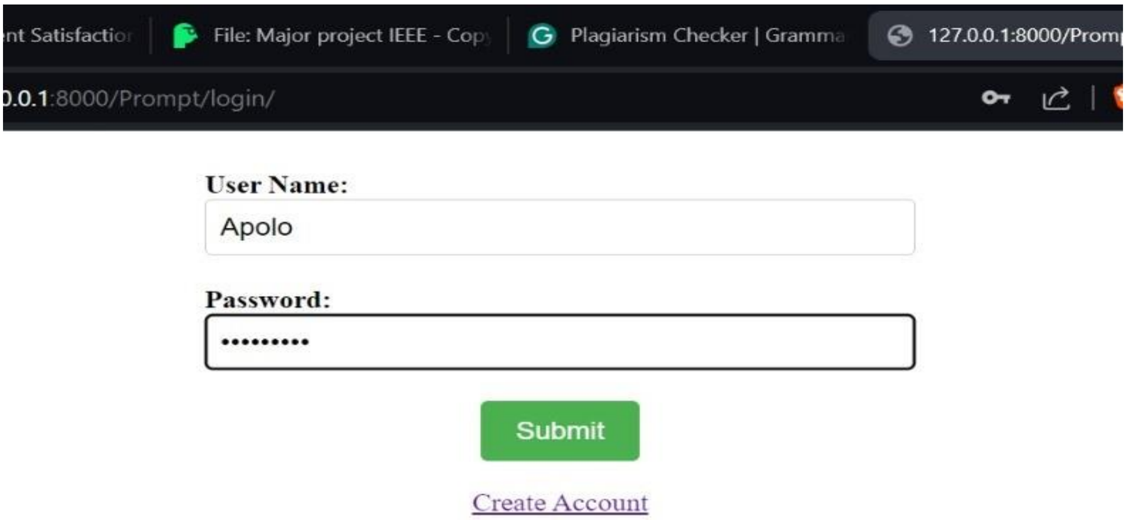
```
def index(request):
    if request.method == 'POST':
        form = MyForm(request.POST)
        if form.is_valid():
            message = form.cleaned_data['message']
            return predict(message, request)
    else:
        form = MyForm()
    return render(request, 'Prompt/prompt_home.html', {'form': form})



def store_admin(request, username):
    address = user_details[username][0]
    if username not in medicine_store:
        medicine_store[username] = set({})
        dump(medicine_store, medicine_store_path)
    user_details[username][1] = medicine_store[username]
    medicines = user_details[username][1]
    context = {
        'username': username,
        'address':address,
        'medicines':medicine_store[username]
    }
    return render(request, 'Prompt/store_admin.html', context)



def login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['UserName']
            password = form.cleaned_data['Password']
            if username in login_cred and (password == login_cred[username]):
                return redirect('store_admin', username=username)
            elif username in login_cred:
```

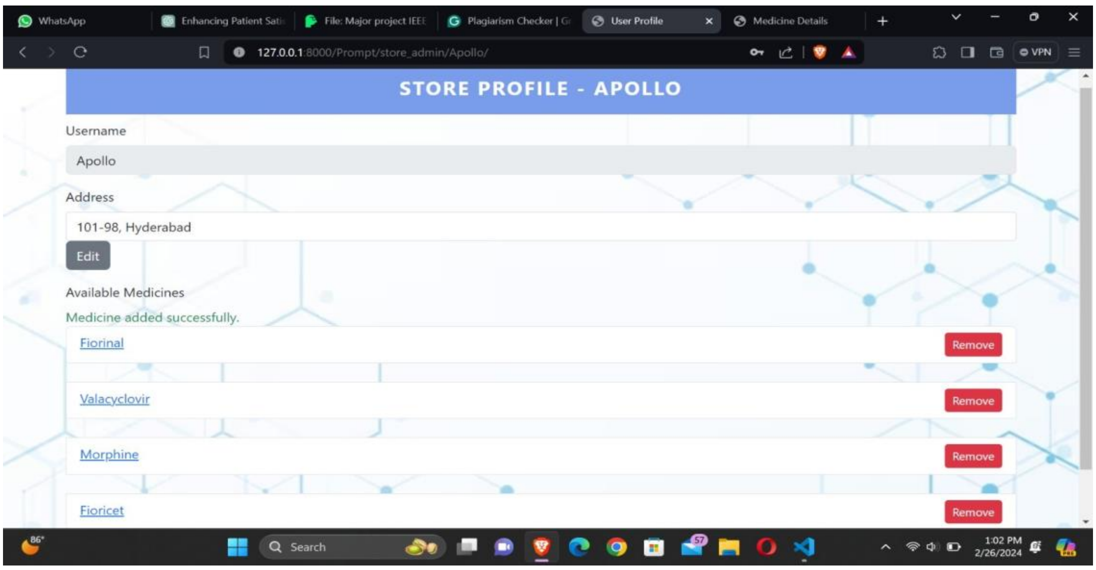
```
                messages.error(request, 'Incorrect password.')
            else:
                messages.error(request, 'Invalid Username, Create account?')
        else:
            messages.error(request, 'Invalid username or password.')
    else:
        form = MyForm()
    return render(request, 'Prompt/login.html', {'form': form})



def medicine_details(request):
    name = request.GET.get('name')
    availability = []
    for i in list(medicine_store.keys()):
        if name in medicine_store[i]:
            availability.append([i,user_details[i][0]])
    if name in description:
        descr = description[name]
    else:
        descr = 'Description not available'
    return render(request, 'Prompt/medicine_details.html', {'description': descr,
'availability':availability })


def create(request):
    if request.method == 'POST':
        form = CreateAccountForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['UserName']
            password = form.cleaned_data['Password']
            reenter_password = form.cleaned_data['ReenterPassword']
            if username not in login_cred:
                if password != reenter_password:
                    messages.error(request, 'Passwords do not match.')
                else:
                    strong, message = check_password_strength(password)
```

```
                if strong:
                    login_cred[username] = password
                    messages.error(
                        request, 'Account created successfully, Login into your account!')
                    user_details[username] = ['', {}]
                    dump(user_details, user_details_path)
                    dump(login_cred, file_path)
                    return redirect('login')
                else:
                    messages.error(request, message)
            else:
                messages.error(
                    request, 'Username {} already taken!'.format(username))

        else:
            for field, errors in form.errors.items():
                for error in errors:
                    messages.error(request, f'{field}: {error}')
    else:
        form = CreateAccountForm()
    return render(request, 'Prompt/create_account.html', {'form': form})



def check_password_strength(password):
    if len(password) < 8:
        return False, "Password must be at least 8 characters long"

    if not re.search("[a-z]", password) or not re.search("[A-Z]", password):
        return False, "Password must contain both uppercase and lowercase letters"

    if not re.search("[0-9]", password):
        return False, "Password must contain at least one digit"

    if not re.search("[ !\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~]", password):
        return False, "Password must contain at least one special character"
```

```
    return True, "Password is strong"



from django.http import JsonResponse
def save_address(request):
    if request.method == 'POST':
        address = request.POST.get('address')
        username = request.POST.get('username')
        user_details[username][0] = address
        dump(user_details, user_details_path)
        return JsonResponse({'message': 'Address for {} has been saved'.format(username)})
    return JsonResponse({'error': 'Invalid request'})

import json

def update_medicine(request):
    if request.method == 'POST':
        data = json.loads(request.body.decode('utf-8'))
        medicine = data.get('medicineName')
        username = data.get('username')
        if medicine in medicine_set:
            if medicine not in medicine_store[username]:
                if username in medicine_store:
                    medicine_store[username].add(medicine)
                else:
                    medicine_store[username] = set({medicine})


                dump(medicine_store, medicine_store_path)
                return JsonResponse({ 'status': 'Success', 'message': 'Medicine added .' })
            else:
                return JsonResponse({ 'status': 'Error', 'message': 'Medicine {} already
available'.format(medicine) })
        else:
            return JsonResponse({ 'status': 'Error', 'message': 'Unknown Medicine
```

```
{}'.format(medicine) })
    return JsonResponse({ 'status': 'Error', 'message': 'Failed to update medicine.' })


def remove_medicine(request):
    if request.method == 'POST':
        data = json.loads(request.body)
        item_name = data.get('itemName')
        username = data.get('userName')
        medicine_store[username].remove(item_name)
        return JsonResponse({ 'status': 'Success', 'message': 'Medicine removed successfully!'
})
    else:
        return JsonResponse({ 'status': 'Error', 'message': 'Failed to remove medicine details!' })
```





url.py:

```
from django.urls import path
from . import views

urlpatterns = [
    path("", views.index, name="index"),
    path('login/', views.login, name='login'),
    path('create/', views.create, name='create'),
    path('save_address/', views.save_address, name='save_address'),
    path('update_medicine/', views.update_medicine, name='update_medicine'),
    path('store_admin/<str:username>/', views.store_admin, name='store_admin'),
    path('remove_medicine', views.remove_medicine, name='remove_medicine'),
    path('medicine_details/', views.medicine_details, name='medicine_details')
]
```

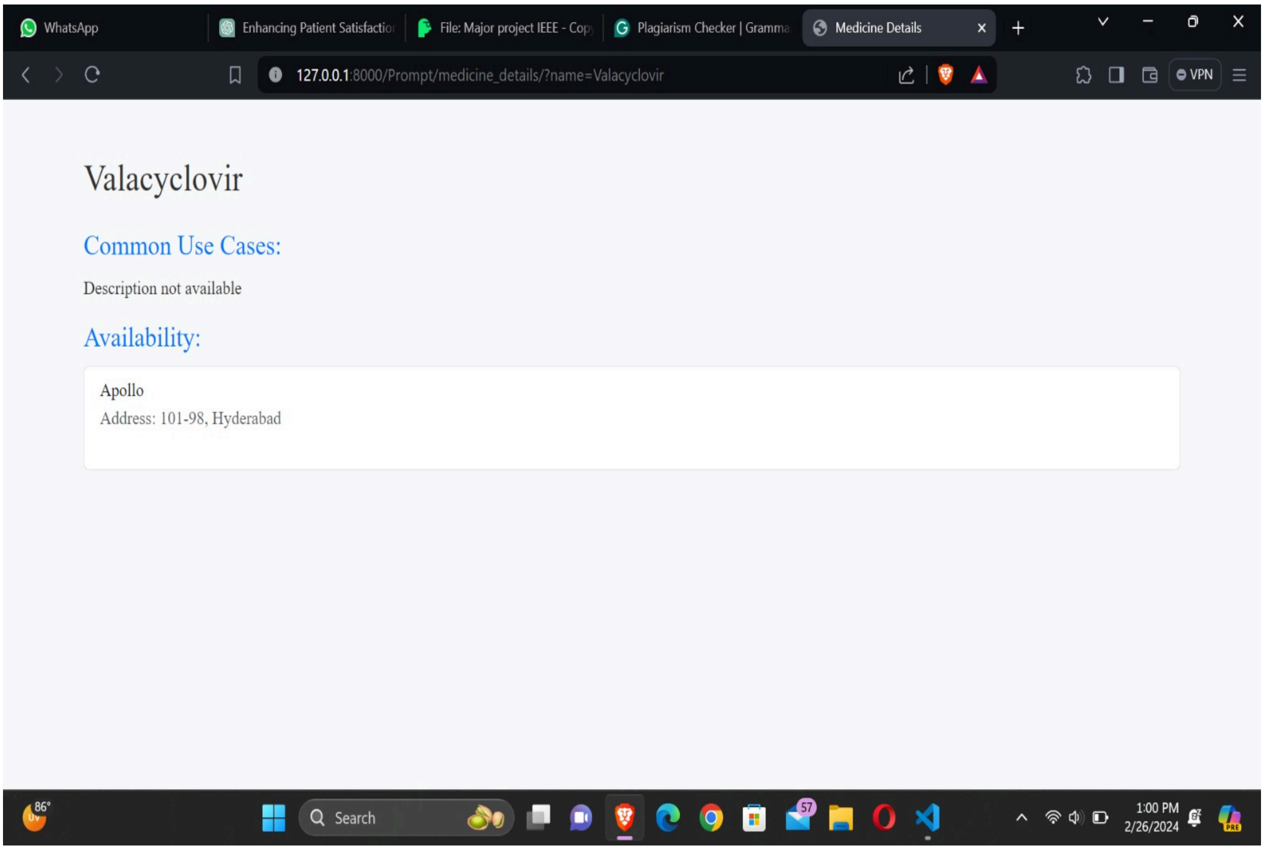
settings.py:

```
from pathlib import Path
import os

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = 'django-insecure-x7%ubsfmw$imgl1o6a5sknc-5uth)f2g#wlppib5(kbn$dja*7'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
```

```
    'Prompt'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'majorproject.urls'

STATICFILES_DIRS = [os.path.join(BASE_DIR,'myapp\static'),]



TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'Prompt', 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
WSGI_APPLICATION = 'majorproject.wsgi.application'
```

```
# Database
# https://docs.djangoproject.com/en/5.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/5.0/topics/i18n/

LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```



CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

The exploration of existing solutions sheds light on the diverse approaches and methodologies available to enhance the capabilities of integration of multiple models.

A Computer-based disease prediction and medicine recommendation system using machine learning approach

Approach:

The computer based disease approach is a disease prediction and medication recommendation using machine learning algorithms: Decision Tree Classifier, Random Forest Classifier, and Naive Bayes Classifier. Accurate disease prediction is crucial for effective treatment, and the system aims to predict diseases based on patient-reported symptoms, subsequently recommending tailored medications. The study emphasizes the importance of accurate prediction in healthcare, highlighting the detrimental effects of erroneous predictions. By combining insights from multiple classifiers, the system enhances prediction accuracy and reliability, ultimately improving patient outcomes. Overall, the paper underscores the significance of accurate disease prediction and medication recommendation in modern healthcare and demonstrates the efficacy of the proposed system through rigorous experimentation.

Benefits:

The use of machine learning algorithms enables accurate prediction of diseases based on symptoms, enhancing the potential for correct treatment. Combining predictions from multiple classifiers (Decision Tree, Random Forest, Naive Bayes) improves prediction accuracy and reduces the risk of misdiagnosis. The system not only predicts diseases but also recommends appropriate medications based on the predicted disease, enhancing the overall treatment process.

Comparison:

The accuracy of disease prediction heavily relies on the quality and completeness of the input data, which may vary and affect the reliability of predictions. The system's effectiveness may be limited to the diseases and symptoms included in the dataset, potentially missing out on rare or emerging diseases not covered in the training data. Machine learning models like Decision Trees and Random Forests may lack interpretability, making it challenging to understand the underlying reasoning behind predictions. Accordingly in our project we mostly depended on the store manager, who gets to upload the info. As the products get into the store the intern who is available in the store will update the medicine regularly. so we don't arise the data issues.

Drug Recommendation System for Diabetes Using a Collaborative Filtering and Clustering Approach

Approach:

The critical need for accurate disease prediction and medication recommendation in healthcare, particularly in the context of increasing patient data complexity. Leveraging machine learning algorithms like Decision Tree, Random Forest, and Naive Bayes classifiers, the study aims to develop predictive models for disease prognosis based on patient symptoms. The dataset utilized, sourced from the UCI Machine Learning Repository, contains comprehensive patient records spanning a decade, primarily focusing on diabetes-related diseases. By employing ensemble learning techniques and robust model evaluation metrics, the study endeavors to enhance prediction accuracy and facilitate personalized treatment strategies. Ultimately, the introduction sets the stage for exploring how machine learning can revolutionize disease prediction and medication recommendation, ultimately improving patient outcomes in healthcare delivery.

Benefits:

The dataset provides a detailed overview of patient health records, enabling in-depth analysis and exploration of patterns and trends. And also EDA helps in understanding the dataset, identifying patterns, anomalies, and testing hypotheses, thereby laying a strong foundation for subsequent analysis.

Comparison:

In the drug recommendation system for diabetes using a collaborative filtering and clustering approach is most likely similar to our project. But this project helps for only the patients who are suffering from diabetes. Our aim is to provide drugs for as possible patients as we can according to their diseases.

A Machine Learning based Drug Recommendation System for Health Care

Approach:

The Drug Recommendation system for Health Care approach is based on Studies, such as the Pew Internet survey, reveal that 55% of internet users seek health-related information online. Researchers have also analyzed search terms entered search engines to understand common health-related queries. Additionally, according to a report from NCBI, around 99,000 people die annually due to mistakes made by medical professionals in hospitals. Recommendation systems, leveraging technologies like Machine Learning and Data Mining, are crucial in today's rapidly advancing technological landscape, as they can potentially save lives. The system employs advanced technologies such as Machine Learning and Data Mining, along with a Content and Collaborative filtering approach. Its primary objective is to design an effective and accurate system for recommending drugs to patients. With the abundance of data available on the internet, our system aims to analyze this data accurately, efficiently, and at scale to fulfill its objective.

Benefits:

The proposed drug recommendation system enhances access to health-related information, allowing users to make informed decisions about medication choices based on patient reviews and ratings. By leveraging technologies like Machine Learning and Data Mining, the system provides personalized medication recommendations tailored to individual health conditions and preferences. Recommendation systems in healthcare have the potential to save lives by reducing medication errors and improving the accuracy of treatment decisions.

Comparison:

The accuracy and reliability of patient reviews and ratings may vary, potentially leading to incorrect recommendations or biases in the system. Collecting and analyzing personal health data raises privacy concerns, necessitating stringent measures to protect user confidentiality and comply with data protection regulations. The effectiveness of the recommendation system relies heavily on the performance and accuracy of the underlying technologies such as Machine Learning and Data Mining. Any inaccuracies or limitations in these technologies could impact the reliability of the recommendations. As comparatively our project might goes through this above situations. But our scope is to overcome the circumstances.

CHAPTER 5
CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 5

CONCLUSION

The Drug Recommender System utilizes machine learning for sentiment analysis, a significant advancement in healthcare technology. This innovative approach helps medical professionals and patients make informed drug related decisions by extracting and categorizing attitudes in medication evaluations. The sentiment-aware drug recommendation engine prioritizes medications with high sentiment ratings and addresses issues in unfavorable reviews, resulting in personalized treatment programs. The Drug Recommender System is effective in making trustworthy medicine recommendations, improving patient outcomes and satisfaction.

Its modular architecture and scalability enable easy maintenance and upgrades. Future improvements include deep learning methods, multimodal sentiment analysis, and genetic characteristics, enhancing its capabilities and providing more personalized prescription recommendations.

This system aims to transform the healthcare industry by providing doctors with evidence- based decision-making tools and patients with treatments tailored to their individual needs and health profiles. The Drug Recommender System utilizes machine learning for sentiment analysis to improve medical procedures and patient outcomes. This technology fosters a patient-centric, data-driven healthcare ecosystem by utilizing sentiment analysis, advanced algorithms, and a user-centric approach. It offers smarter, individualized, and successful prescription recommendations for global healthcare advancements in technology and medical understanding. Comments play an essential role in our daily decision-making processes, influencing choices such as online shopping or dining out. To assist in these decisions, we often turn to reviews for guidance.

This study employs multiple machine learning classifiers, including logistic regression, perceptron, polynomial naive Bayes, ridge classifier, stochastic gradient descent, linear SVC, Arc reference, TF-IDF, trimmed trees, and random forests, to develop a consensus model. Word2Vec and manual methods are utilized, with LGBM and CatBoost serving as the classifiers for these techniques.Evaluation of these models is conducted using precision, recall, F1-score, accuracy, and AUC score metrics. Results indicate that TF-IDF paired with linear SVC achieves the highest accuracy at 91.5% compared to other models tested.

FUTURE ENHANCEMENT

Future research will focus on comparing different methodologies, exploring alternative n-gram approaches, and devising strategies to further enhance recommendation system performance. This study aims to contribute to the advancement of recommendation systems, ultimately aiding users in making informed decisions across various domains.

REFERENCES

REFERENCES

1. Xiaojiang Lei, Xueming Qian, Member, IEEE, and Guoshuai Zhao, “Rating Prediction Based on Social Sentiment from Textual Reviews”, IEEE transactions on multimedia ,Vol. 18,No. 9, September 2016.

2. A. Ng, A. Zheng, and M. Jordan, “Stable algorithms for link analysis,” in Proc. SIGIR Conf. Inform. Retrieval, New Orleans, Louisiana LO, USA, 2001, pp. 258–266.

3. B. Taskar, M. Wong, P. Abbeel, and D. Koller, “Link prediction in relational data,” in Proc. Adv. NeuralInform. Process. Syst., Vancouver, B.C. Canada, 2003.

4. C. Corley, D. Cook, A. Mikler, and K. Singh, “Text and structural data mining of influenza mentions in web and social media,”Int. J. Environ. Res. Public Health, vol. 7, pp. 596– 615, Feb. 2010.

5. D. Liben-Nowell and J. M. Kleinberg, “The link prediction problem for social networks,”J. Am. Soc. Inform. Sci. Technol., vol. 57, pp. 556–559, May 2007.

6. L. Dunbrack, “Pharma 2.0 – social media and pharmaceutical sales and marketing,” inProc. Health Ind. Insights, 2010

7. Rennie, J.D., et al.: Tackling the poor assumptions of naive bayes text classifiers. In: Machine Learning-International Workshop then Conference, vol. 20(2) (2003).

8. Vivek Narayana, Ishan Arora, Arjun Bhatia, “Fast and Accurate Sentiment Classification Using an Enhanced Naïve Bayes Model”, International Conference on Intelligent Data Engineering and Automated Learning.

9. R. S a l a k h u t d i n o v and A. M n ih , “Probabilistic matrix factorization,” in Proc. NIPS, 2007, pp. 1257–1264.

10. X. Yang, H. Steck, and Y. Liu, “Circle- based recommendation in online social networks,” in Proc. 18th ACM SIGKDD Int. Conf. KDD, Aug. 2012, pp. 1267–1275.

11. M. Jiang et al., “Social contextual recommendation,” in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage,2012, pp. 45– 54.

12. M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation insocial networks,” in Proc. ACM Conf.Recommender Syst., 2010, pp. 135–142.

13. Z. Fu et al., “Achieving efficient cloud search services: Multikeyword ranked search over encryptedcloud data supporting parallel computing,” IEICE Trans. Commun., vol. 98, no. 1, pp. 190–200, 2015

14. G. Ganu, N. Elhadad, and A Marian, “ Beyond the stars: Imp rovin g rating predictions using Reviewtext content,” in Proc. 12th Int. Workshop Web Databases, 2009, pp. 1–6.

15. J. Xu, X. Zheng, and W. Ding, “Personalized recommendation based on reviews and ratings alleviating the sparsity problem of collaborative filtering,” in Proc. IEEE 9th Int. Conf. e-Bus. Eng., Sep. 2012, pp. 9–16.

16. X. Qian, H. Feng, G. Zhao, and T. Mei, “Personalized recommendation combining user interest and social circle,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 7, pp. 1763–1777, Jul. 2014.

GitHub Link:

https://github.com/tharun220103/major-project-B04

DOI:

https://doi.org/10.22214/ijraset.2024.59247