A Project Report on

Disease condition predictor using AI

Submitted in partial fulfillment of the requirements for the award of the degree of
Bachelor of Engineering - Information Technology
by

Mohammed Abdul Rasheed	160321737032
Abdul Rahman	160321737031
Abdu Rahman	160321737046

Under the Guidance of

Ms. Juveria Azeem

Assistant Professor
Dept. of Information Technology DCET, Hyderabad.

DEPARTMENT OF INFORMATION TECHNOLOGY
DECCAN COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to Osmania University)

Hyderabad

2023-2024

Deccan College of Engineering & Technology

(AFFILIATED TO OSMANIA UNIVERSITY)

Estd. By DAR-US -SALAM EDUCATIONAL TRUST

Dar-us-salaam, Near Nampally, Hyderabad-500 001.

CERTIFICATE

This is to certify that the project report entitled **Disease Condition Predictor Using AI** being submitted by **Mr. Mohammed Abdul Rasheed (160321737032), Mr. Abdul Rahman (160321737031), Mr. Abdu Rahman (160321737046),**

in partial fulfillment for the award of the Degree of Bachelor of Engineering in

Information Technology by the Osmania University is a record of bonafide work carried out by them under my guidance and supervision.

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

Internal Guide	**Project Coordinator**	**HOD**	**External Examiner**
Ms. Juveria Azeem	**Ms. Shaheen Begum**	**Dr. Ayesha Ameen**	
Associate Professor	**Associate Professor**	**Professor and Head**	
Dept. Of IT	**Dept. Of IT**	**Dept. Of IT**	
DCET, Hyderabad.	**DCET, Hyderabad**	**DCET, Hyderabad.**	

Declaration

This is to certify that the work reported in the present project entitled

"Disease Condition Predictor using AI"

is a record of work done by me in the Department of Information Technology, Deccan College of Engineering and Technology, Osmania University, Hyderabad in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering information Technology.

The results presented in this dissertation have been verified and are found to be satisfactory. The results embodied in this dissertation have not been submitted to any other university for the award of any degree or diploma.

Mohammed Abdul Rasheed (160321737032)

Abdul Rahman (160321737031)

Abdu Rahman (160321737046)

Acknowledgement

I would like to express my sincere gratitude and indebtedness to my project supervisor.

Ms. Juveria Azeem for her valuable suggestions and interest throughout the course of the project.

I am thankful to Head of Department **Dr. Ayesha Ameen** for providing excellent infrastructure and a nice atmosphere for complementing this project.

I convey my heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed.

Finally, I would like to take this opportunity to thanks my family for their support through the work. I sincerely acknowledge and thanks who gave directly or indirectly their support in this project work.

Mohammad Abdul Rasheed (160321737032)

Abdul Rahman (160321737031)

Abdu Rahman (160321737046)

Abstract

The "Disease-Condition-Predictor Using AI" project aims to leverage Natural Language Processing (NLP) and Machine Learning (ML) techniques to predict disease conditions based on patient reviews of specific drugs. By analyzing textual data, the project provides a framework for personalized healthcare recommendations, enhancing patient care through AI-driven insights.

The project utilizes a dataset containing drug reviews, associated conditions, and patient satisfaction ratings. The workflow is divided into six key phases: Data Cleaning, Exploratory Data Analysis (EDA), Text Pre-processing, Model Building, Result Visualization, and Web App Development. Data cleaning involved handling missing values, removing duplicates, and formatting columns. EDA provided insights into the most common drugs, conditions, and rating distributions. Text pre-processing included tokenization, lemmatization, and corpus building to prepare the data for model training.

The core of the project lies in the model-building phase, where Term Frequency-Inverse Document Frequency (TF-IDF) was applied to transform text data into numerical features. Classification algorithms such as Logistic Regression, Support Vector Classifier (SVC), Random Forest, and Decision Trees were employed to predict the disease conditions.

Finally, a web application was developed using Streamlit to allow users to input text and receive predictions on potential disease conditions based on the trained SVC model. This project provides a practical application of AI in healthcare, showcasing the potential for automating disease prediction and enhancing patient outcomes through personalized recommendations.

Contents

1. Introduction

1.1 What is Disease condition Predictor?

The **Disease Condition Predictor** is an AI-powered system developed to assist healthcare professionals, especially doctors, in accurately diagnosing conditions and prescribing appropriate medications based on patient-reported data. Leveraging advanced Natural Language Processing (NLP) and Machine Learning (ML) techniques, the predictor analyzes textual data from patient reviews, symptoms, and feedback related to specific drugs. This analysis helps identify correlations between the language patients use to describe their conditions and the underlying diseases they may be experiencing.

The primary goal of the Disease Condition Predictor is to streamline the process of understanding patient conditions and recommending suitable treatments. By offering AI-driven insights, the model helps doctors decipher large volumes of unstructured data, allowing them to recognize patterns and make evidence-based decisions more efficiently. This not only enhances diagnostic accuracy but also aids in personalizing drug prescriptions based on the nuances of individual patient experiences.

In practice, the system enables doctors to gain a deeper understanding of how patients respond to particular drugs and how their reviews may indicate specific health conditions. This empowers healthcare providers to make informed choices tailored to the unique needs of each patient, improving overall treatment outcomes. Furthermore, the model helps bridge the gap between patient language and medical terminology, offering a valuable tool for enhancing patient-doctor communication and reducing diagnostic errors.

1.2 Applications

1. **Enhanced Diagnosis:**

- **Early Detection:** Helps identify potential disease conditions early by analyzing patient reviews and feedback, enabling timely intervention and treatment.
- **Accurate Diagnosis:** Assists doctors in correlating symptoms described by patients with known medical conditions, leading to more accurate diagnoses.

2. **Personalized Drug Prescription:**

- **Tailored Treatments:** Provides insights into how different patients respond to medications, allowing doctors to prescribe drugs that are more likely to be effective for individual cases.
- **Adjusting Dosages:** Helps in adjusting medication dosages based on patient-reported experiences and outcomes.

3. **Improved Patient Care:**

- **Understanding Patient Conditions:** Enables a better understanding of patient conditions by analyzing their descriptions and feedback, leading to more empathetic and informed care.
- **Monitoring Treatment Efficacy:** Assists in monitoring how well treatments are working by analyzing patient feedback and adjusting treatment plans as needed.

4. **Streamlined Healthcare Decision-Making:**

- **Data-Driven Insights:** Provides doctors with data-driven insights from patient reviews, facilitating quicker and more informed decision-making.
- **Reducing Diagnostic Errors:** Helps in reducing diagnostic errors by cross-referencing patient feedback with medical conditions and treatment responses.

5. **Enhanced Patient-Doctor Communication:**

- **Clarifying Symptoms:** Assists in translating patient-reported symptoms into medical terminology, improving communication between patients and healthcare providers.
- **Educational Tool:** Serves as an educational tool for doctors to better understand the common issues and concerns expressed by patients in their reviews.

6. **Healthcare Research and Development:**

- **Data Analysis:** Supports research by providing a large dataset of patient reviews and drug responses for further analysis and study.
- **Drug Development:** Assists pharmaceutical companies in understanding patient feedback on existing drugs, aiding in the development of new treatments.

7. Web-Based Patient Interaction:

- **User-Friendly Interface:** Through a web app, allows patients to input their symptoms and receive preliminary insights or recommendations, enhancing patient engagement and self-management.

2. Literature Survey

 Predicting Disease Outcomes with Machine Learning
Authors: Miotto, R., Wang, F., Wang, S., et al. (2016)

Summary: This study presents Deep Patient, a deep learning model designed to predict patient outcomes based on electronic health records (EHRs). The model demonstrates the efficacy of using deep learning to analyze complex patient data, highlighting how predictive models can enhance disease prognosis. The findings underscore the potential of machine learning in predicting health conditions and improving patient care.

 Sentiment Analysis of Healthcare Data: A Systematic Review
Authors: Liu, B. (2018)

Summary: This review paper provides a comprehensive overview of sentiment analysis applied to healthcare data, including patient reviews and feedback. It discusses various methodologies used to extract meaningful insights from textual data, emphasizing the relevance of sentiment analysis in understanding patient experiences and predicting healthcare outcomes.

 Leveraging Patient Reviews for Disease Prediction and Management
Authors: Wang, H., Zhang, S., Wang, Y., et al. (2017)

Summary: The paper explores the use of patient reviews to predict disease conditions and manage healthcare. It covers techniques for analyzing textual reviews to identify patterns and correlations with medical conditions, demonstrating the potential of text mining and NLP in improving disease prediction and personalized care.

 Personalized Medicine: A Review of the Progress and Current Trends
Authors: Collins, F. S., Varmus, H. (2015)

Summary: This review discusses the advancements in personalized medicine, including the integration of data-driven approaches for tailoring treatments to individual patients. It highlights the role of predictive modeling and machine learning in developing personalized healthcare solutions, providing context for the use of AI in drug prescription and disease management.

 Machine Learning Approaches for Drug Discovery and Development
Authors: Chen, J., Song, L., Wang, S., et al. (2018)

Summary: The paper examines how machine learning techniques are applied in drug discovery and development. It focuses on predictive models for drug responses and treatment optimization, relevant to understanding how ML can enhance personalized drug prescriptions based on patient data.

3. System Specification

3.1 System Overview

Purpose : The Disease Condition Predictor is designed to analyze patient reviews and predict disease conditions using natural language processing (NLP) and machine learning (ML) techniques. The system aims to assist healthcare professionals by providing insights into patient conditions and recommending appropriate treatments based on textual feedback

3.2 Hardware Requirements

- **Computer System:** A desktop or laptop with at least an Intel Core i5 CPU, 8 GB RAM, and 250 GB SSD for development and testing.
- **Graphics Processing Unit (GPU):** An integrated GPU is sufficient for basic tasks, but a dedicated GPU (e.g., NVIDIA GeForce GTX) is recommended for training complex models.

3.3 Software Requirements

- **Operating System:**
  Minimum:Windows 10, macOS Mojave, or Ubuntu 20.04 LTS.

- **Development Tools:**
  - **Python:** Programming language for implementing the models and processing data.
  - **Integrated Development Environment (IDE):** VS Code or Jupyter Notebook for coding and experimentation.

- **Libraries and Frameworks:**
  - **Numpy:** For numerical computations.
  - **Pandas:** For data manipulation and analysis.
  - **Matplotlib:** For basic data visualization.
  - **scikit-learn:** For machine learning algorithms and model building.
  - **NLTK:** For natural language processing tasks.
  - **Streamlit:** For creating a web-based interface to interact with the model.

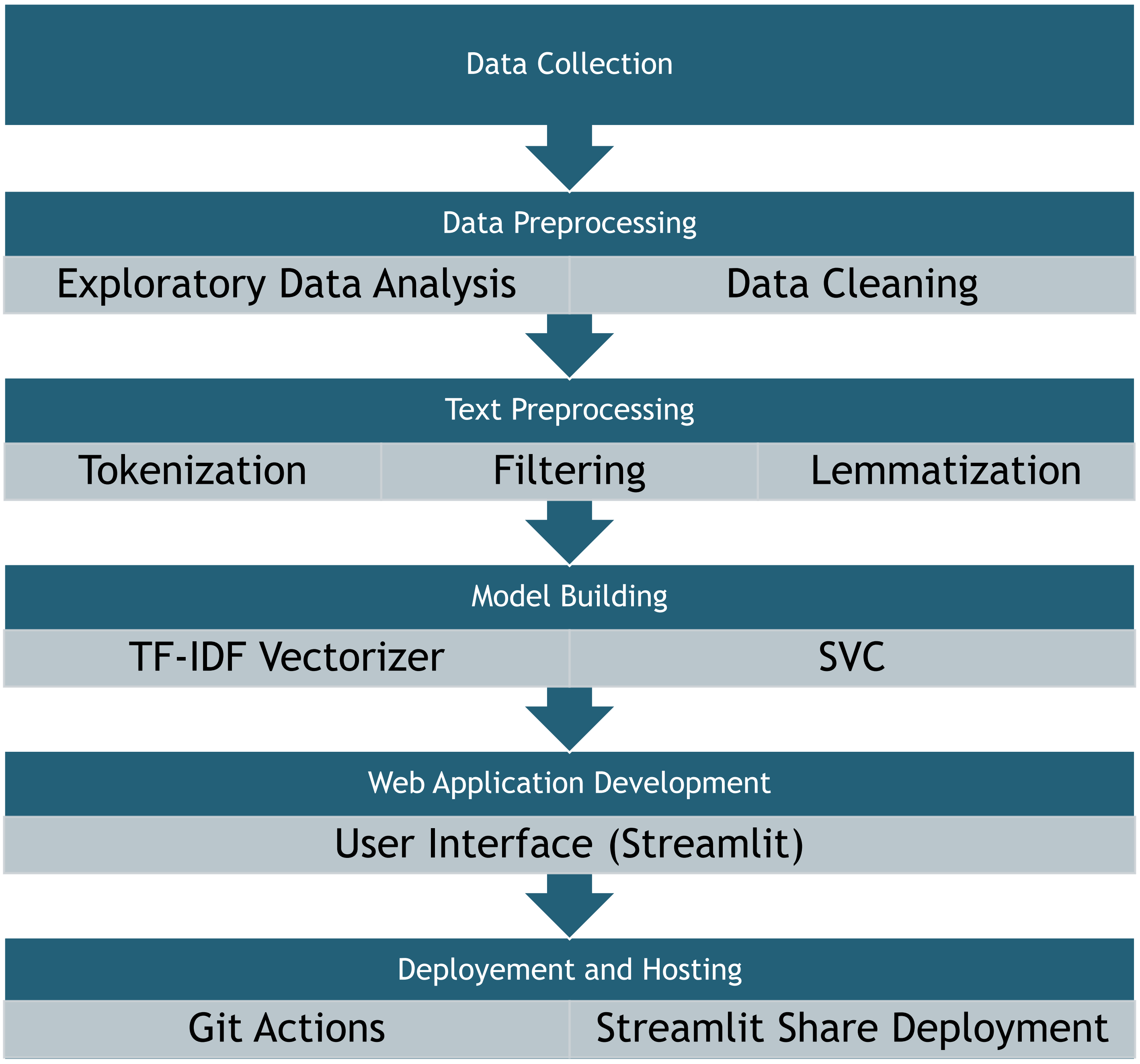
3.4 Objective

The primary objective of the Disease Condition Predictor project is to develop a sophisticated tool that leverages natural language processing (NLP) and machine learning (ML) techniques to analyze patient reviews and predict disease conditions. The key objectives are:

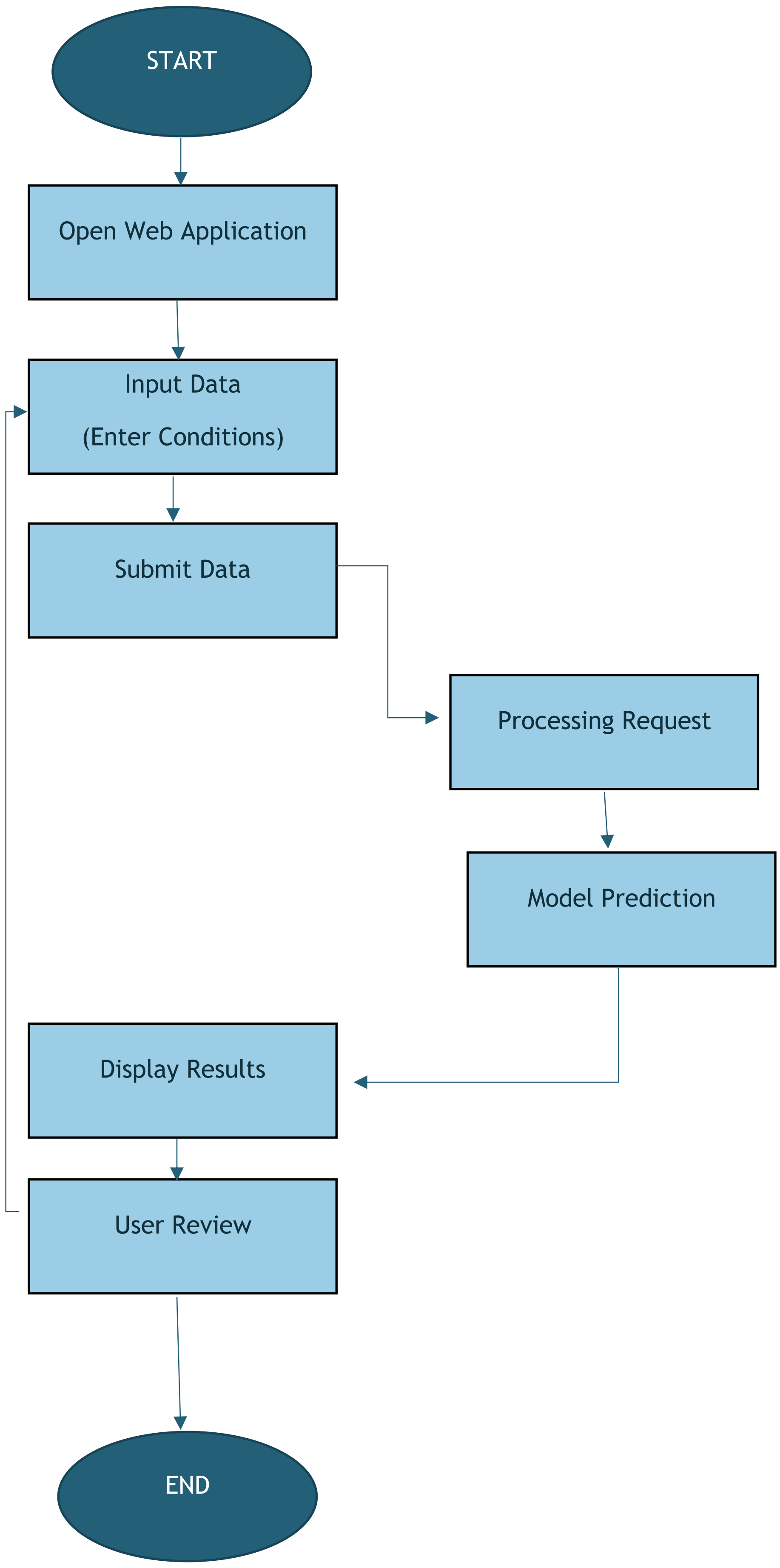
1. **Enhanced Disease Prediction:**
    - To accurately predict disease conditions from textual patient reviews using advanced ML models. This involves extracting meaningful insights from unstructured text data to identify potential health issues.
2. **Improved Healthcare Recommendations:**
    - To provide healthcare professionals with actionable insights and personalized recommendations based on the analysis of patient reviews. This aims to assist doctors in prescribing appropriate treatments and understanding patient conditions more effectively.
3. **Integration of NLP and ML Techniques:**
    - To implement state-of-the-art NLP techniques for text preprocessing, feature extraction, and sentiment analysis. To apply various ML algorithms, including TF-IDF vectorization and classification models, to build robust predictive models.
4. **User-Friendly Interface:**
    - To develop a web-based application using Streamlit that allows users (e.g., healthcare professionals) to easily input patient review data and receive predictions and recommendations. The interface should be intuitive and accessible for practical use in a clinical setting.
5. **Data-Driven Insights:**
    - To conduct exploratory data analysis (EDA) to identify trends, common conditions, and patient sentiments. This analysis will help in understanding the broader implications of patient feedback on disease prediction and management.
6. **Testing and Validation:**
    - To ensure the reliability and accuracy of the predictive models through rigorous testing and validation. This includes evaluating the performance of different algorithms and refining the models based on evaluation metrics.

4. System Design

4.1 System Architecture

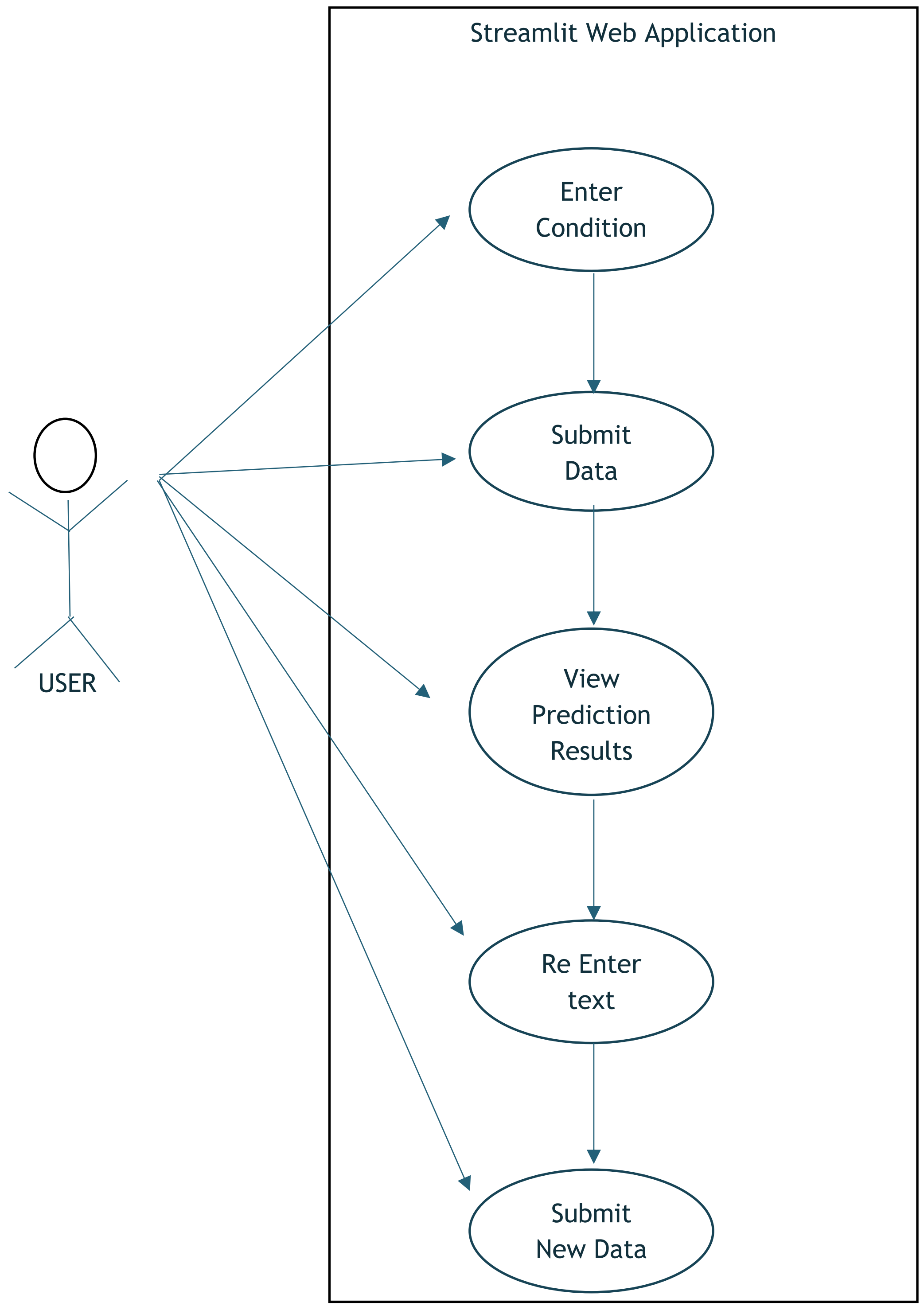


4.2 Flow Chart

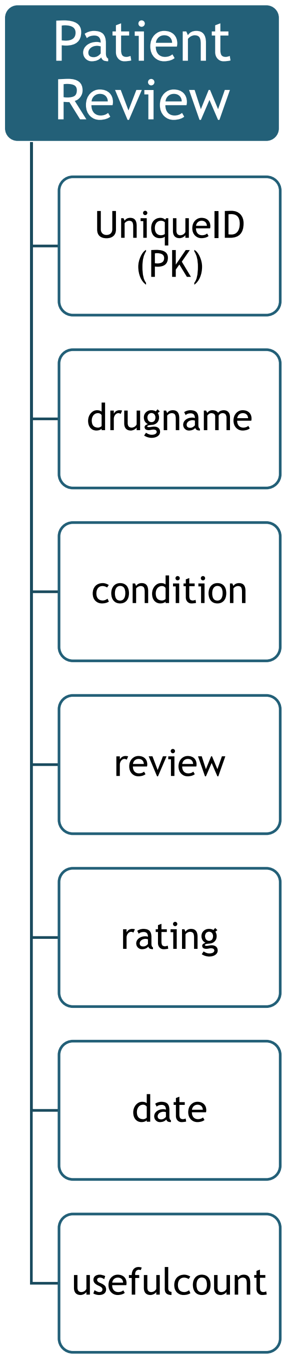


4.3 UML Diagrams

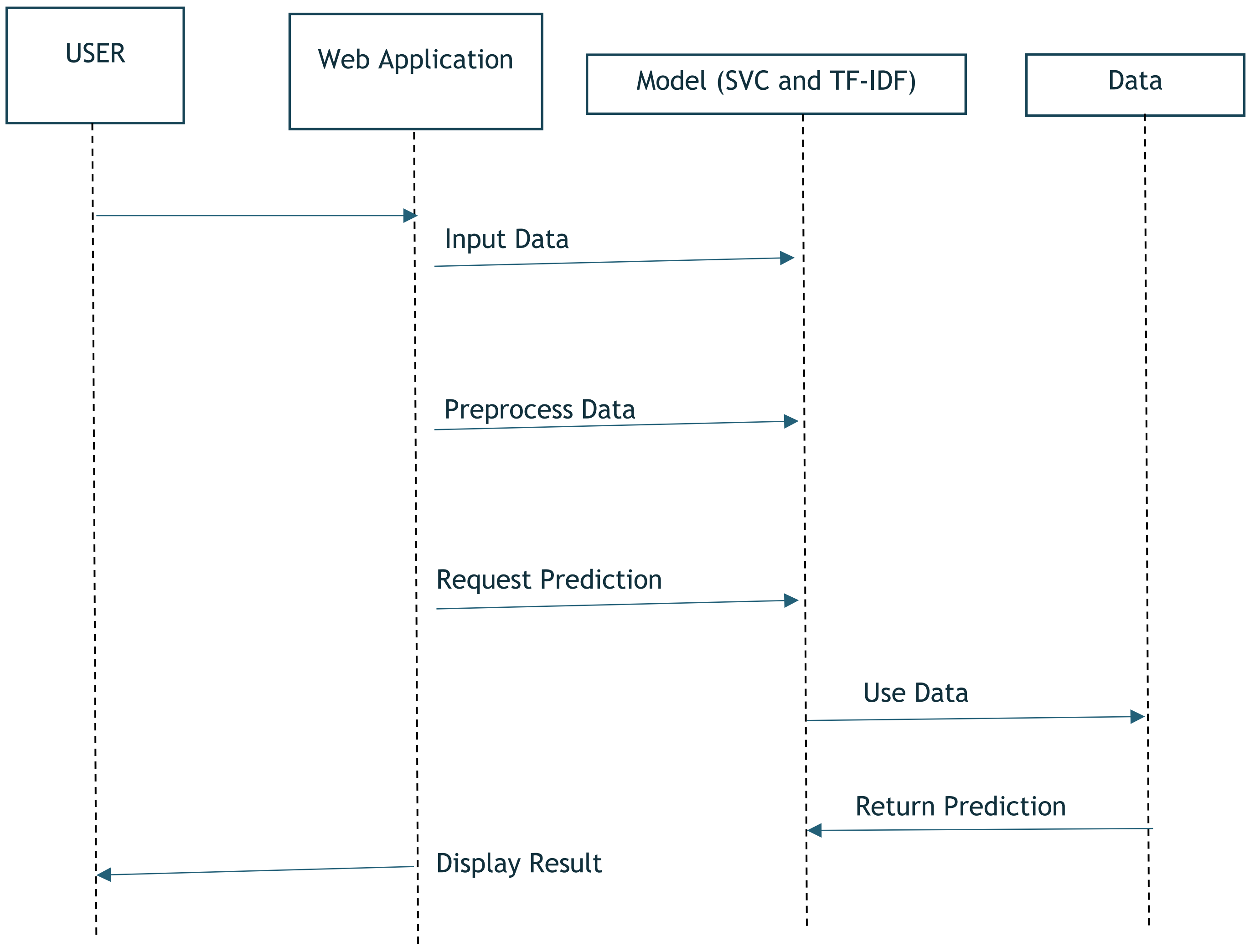
4.3.1 Use Case Diagram



4.3.2 ER Diagram



4.3.3 Sequence Diagram



5. Implementation

5.1 Module Description

1. Disease_Condition_Prediction.ipynb

Description: This Jupyter notebook contains the main code for developing the disease condition prediction model. It includes data cleaning, exploratory data analysis (EDA), text preprocessing, model building (using TF-IDF and various classification algorithms), and evaluation of model performance.

2. README.md

Description: The README file provides an overview of the project, including its purpose, the dataset used, and a brief description of the steps involved in the project. It also includes instructions for running the project and dependencies required.

3. SVC.pkl

Description: This file contains the serialized Support Vector Classifier (SVC) model. It is used to make predictions about disease conditions based on the input reviews. The model is trained and saved in this file for later use.

4. UCIdrug_train.rar

Description: This is the compressed dataset file used for training the model. It includes patient reviews, drug names, conditions, ratings, and other relevant information. The dataset is used to train and test the machine learning models.

5. disease_app.py

Description: This Python script contains the code for the web application developed using Streamlit. It provides a user interface where users can input drug reviews, and it displays the predicted disease condition based on the trained model.

6. diseasedf.zip

Description: This is another compressed dataset file, similar to UCIdrug_train.rar, which may contain the same or additional data used for training and testing the models.

7. requirements.txt

Description: This file lists all the Python libraries and their versions required to run the project. It ensures that the project environment is consistent and includes all necessary dependencies for the code to execute properly.

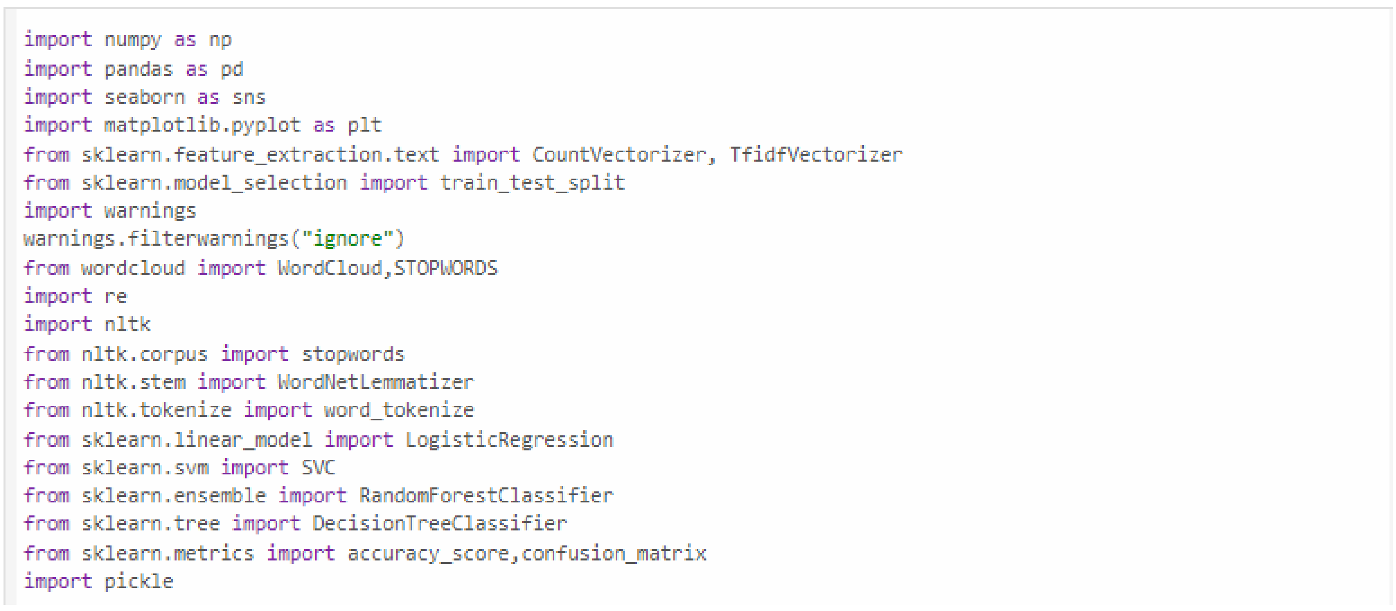
8. vectorizerr.pkl

Description: This file contains the serialized TF-IDF vectorizer used for transforming the text data into numerical format. It is used in conjunction with the trained model to preprocess input data before making predictions.

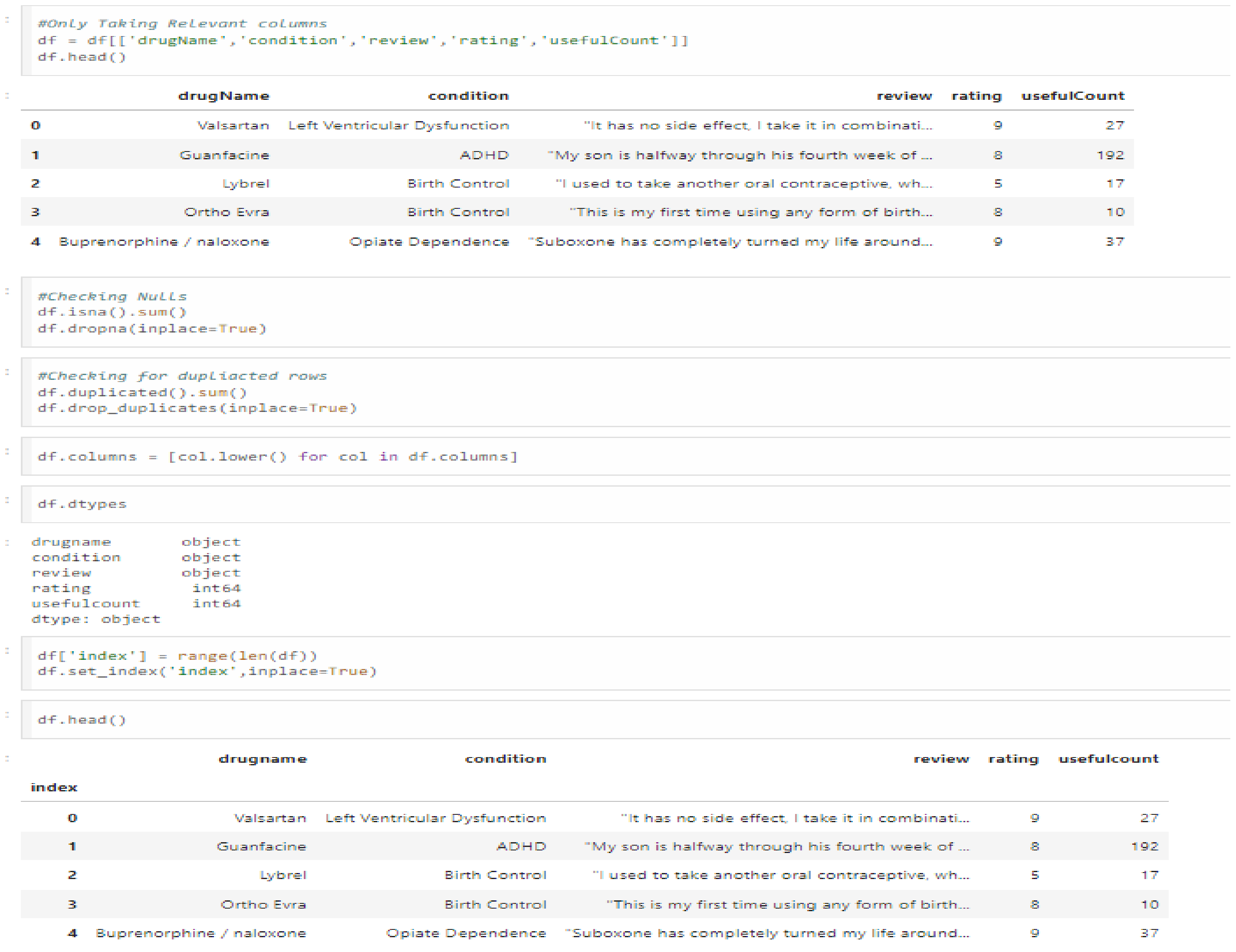
These descriptions provide a clear understanding of the purpose and functionality of each module in your project.

5.2 Code

5.2.1 Data Cleaning and importing necessary libraries



```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings("ignore")
from wordcloud import WordCloud,STOPWORDS
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,confusion_matrix
import pickle
```



```python
#Only Taking Relevant columns
df = df[['drugName','condition','review','rating','usefulCount']]
df.head()
```

	drugName	condition	review	rating	usefulCount
0	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9	27
1	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8	192
2	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5	17
3	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8	10
4	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9	37

```python
#Checking Nulls
df.isna().sum()
df.dropna(inplace=True)
```

```python
#Checking for dupliacted rows
df.duplicated().sum()
df.drop_duplicates(inplace=True)
```

```python
df.columns = [col.lower() for col in df.columns]
```

```python
df.dtypes
```

```
drugname        object
condition       object
review          object
rating           int64
usefulcount      int64
dtype: object
```

```python
df['index'] = range(len(df))
df.set_index('index',inplace=True)
```

```python
df.head()
```

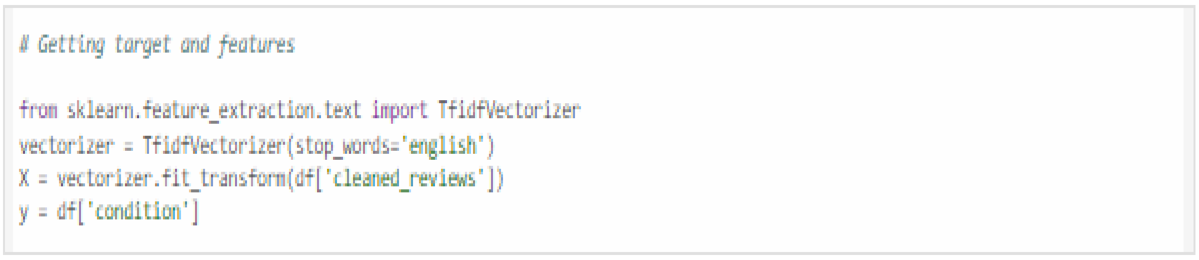
	drugname	condition	review	rating	usefulcount
index					
0	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9	27
1	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8	192
2	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5	17
3	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8	10
4	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9	37

5.2.2 Text Preprocessing

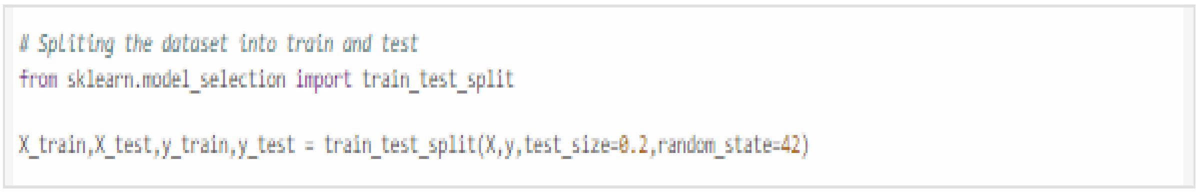
```
In [ ]: len(df['condition'].unique())

Out[ ]: 60

In [ ]: Keep_conditions =[
            "Birth Control",
            "Depression",
            "Body Pain",
            "Anxiety",
            "Acne",
            "Bipolar Disorder",
            "Insomnia",
            "Weight Loss",
            "Obesity",
            "ADHD",
            "Diabetes",
            "Emergency Contraception",
            "High Blood Pressure",
            "Vaginal Yeast Infection",
            "Migraine",'Muscle Spasm','Constipation']
        df = df[df['condition'].isin(Keep_conditions)]
        len(Keep_conditions)

Out[ ]: 17

In [ ]: df['index'] = range(len(df))
        df.set_index('index',inplace=True)

In [ ]: import re
        import nltk
        from nltk.corpus import stopwords
        from nltk.stem import WordNetLemmatizer
        from nltk.tokenize import word_tokenize
        nltk.download('stopwords')
        nltk.download('punkt')
        nltk.download('wordnet')

        [nltk_data] Downloading package stopwords to /root/nltk_data...
        [nltk_data]   Unzipping corpora/stopwords.zip.
        [nltk_data] Downloading package punkt to /root/nltk_data...
        [nltk_data]   Unzipping tokenizers/punkt.zip.
        [nltk_data] Downloading package wordnet to /root/nltk_data...
Out[ ]: True

In [ ]: review = []
        for i in range(len(df)):
          rev = re.sub(r'[^a-zA-z]',' ',df['review'][i])
          rev = rev.lower()
          review.append(rev)

In [ ]: df['cleaned_reviews'] = review

In [ ]: corpus = []
        for i in range(len(df)):
          lemmatizer = WordNetLemmatizer()
          stopword = stopwords.words('english')

          tag = df['cleaned_reviews'][i]

        #Tokenizing
          tag = word_tokenize(tag)

        #Lemmatizing

          tag = [lemmatizer.lemmatize(word) for word in tag if not word in set(stopword)]
          tag  = ' '.join(tag)
          corpus.append(tag)
```

5.2.3 Model Building

5.2.3.1 TF-IDF

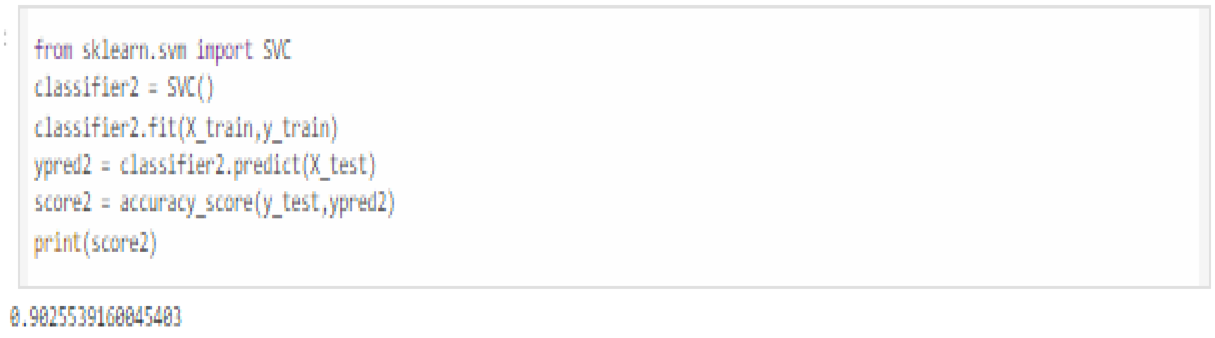


```
# Getting target and features

from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['cleaned_reviews'])
y = df['condition']
```



```
# Spliting the dataset into train and test
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```

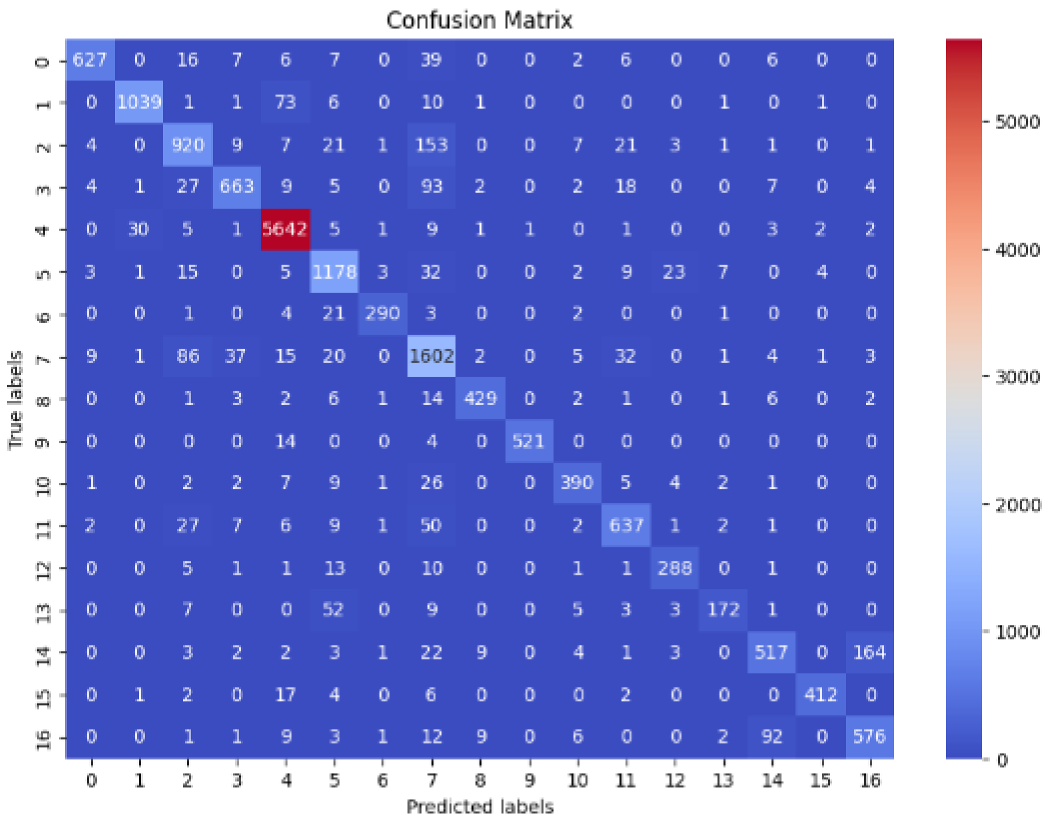
5.2.3.2 Support Vector Classifier (SVC)



```
from sklearn.svm import SVC
classifier2 = SVC()
classifier2.fit(X_train,y_train)
ypred2 = classifier2.predict(X_test)
score2 = accuracy_score(y_test,ypred2)
print(score2)
```

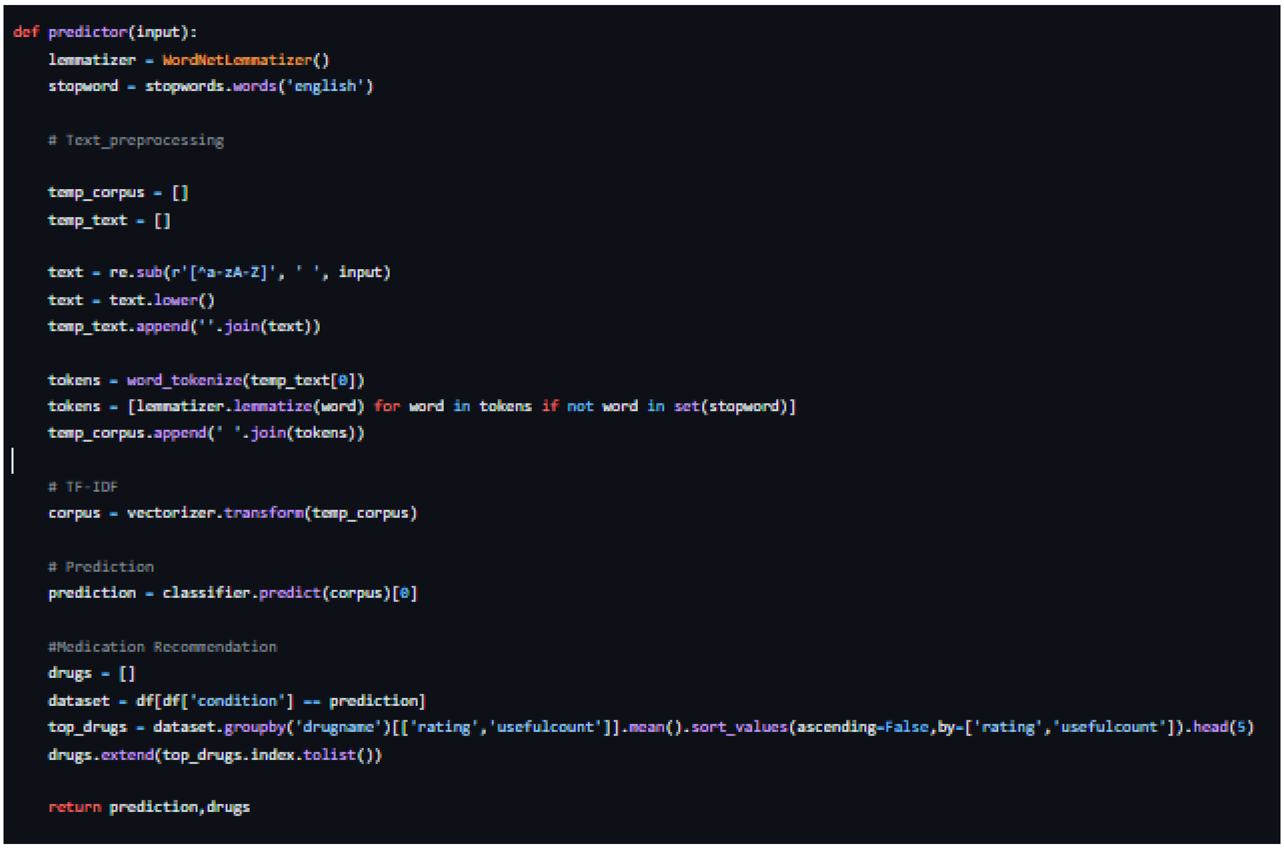
```
0.9025539160045403
```

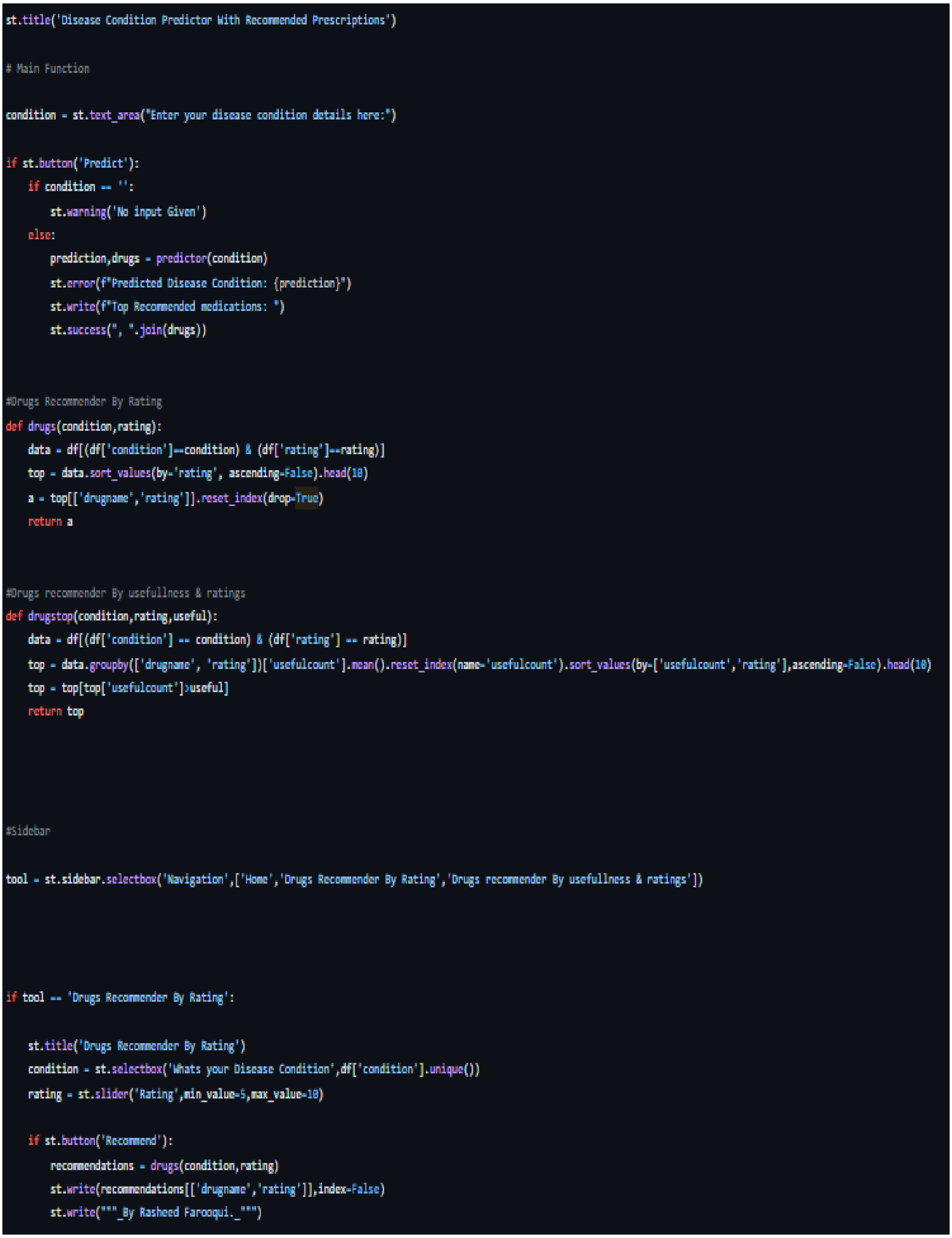
5.2.4 Evaluation


Confusion Matrix

True labels \ Predicted labels	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	627	0	16	7	6	7	0	39	0	0	2	6	0	0	6	0	0
1	0	1039	1	1	73	6	0	10	1	0	0	0	0	1	0	1	0
2	4	0	920	9	7	21	1	153	0	0	7	21	3	1	1	0	1
3	4	1	27	663	9	5	0	93	2	0	2	18	0	0	7	0	4
4	0	30	5	1	5642	5	1	9	1	1	0	1	0	0	3	2	2
5	3	1	15	0	5	1178	3	32	0	0	2	9	23	7	0	4	0
6	0	0	1	0	4	21	290	3	0	0	2	0	0	1	0	0	0
7	9	1	86	37	15	20	0	1602	2	0	5	32	0	1	4	1	3
8	0	0	1	3	2	6	1	14	429	0	2	1	0	1	6	0	2
9	0	0	0	0	14	0	0	4	0	521	0	0	0	0	0	0	0
10	1	0	2	2	7	9	1	26	0	0	390	5	4	2	1	0	0
11	2	0	27	7	6	9	1	50	0	0	2	637	1	2	1	0	0
12	0	0	5	1	1	13	0	10	0	0	1	1	288	0	1	0	0
13	0	0	7	0	0	52	0	9	0	0	5	3	3	172	1	0	0
14	0	0	3	2	2	3	1	22	9	0	4	1	3	0	517	0	164
15	0	1	2	0	17	4	0	6	0	0	0	2	0	0	0	412	0
16	0	0	1	1	9	3	1	12	9	0	6	0	0	2	92	0	576

5.2.5 Web App deployment using streamlit


```python
def predictor(input):
    lemmatizer = WordNetLemmatizer()
    stopword = stopwords.words('english')

    # Text_preprocessing

    temp_corpus = []
    temp_text = []

    text = re.sub(r'[^a-zA-Z]', ' ', input)
    text = text.lower()
    temp_text.append(''.join(text))

    tokens = word_tokenize(temp_text[0])
    tokens = [lemmatizer.lemmatize(word) for word in tokens if not word in set(stopword)]
    temp_corpus.append(' '.join(tokens))

    # TF-IDF
    corpus = vectorizer.transform(temp_corpus)

    # Prediction
    prediction = classifier.predict(corpus)[0]

    #Medication Recommendation
    drugs = []
    dataset = df[df['condition'] == prediction]
    top_drugs = dataset.groupby('drugname')[['rating','usefulcount']].mean().sort_values(ascending=False,by=['rating','usefulcount']).head(5)
    drugs.extend(top_drugs.index.tolist())

    return prediction,drugs
```



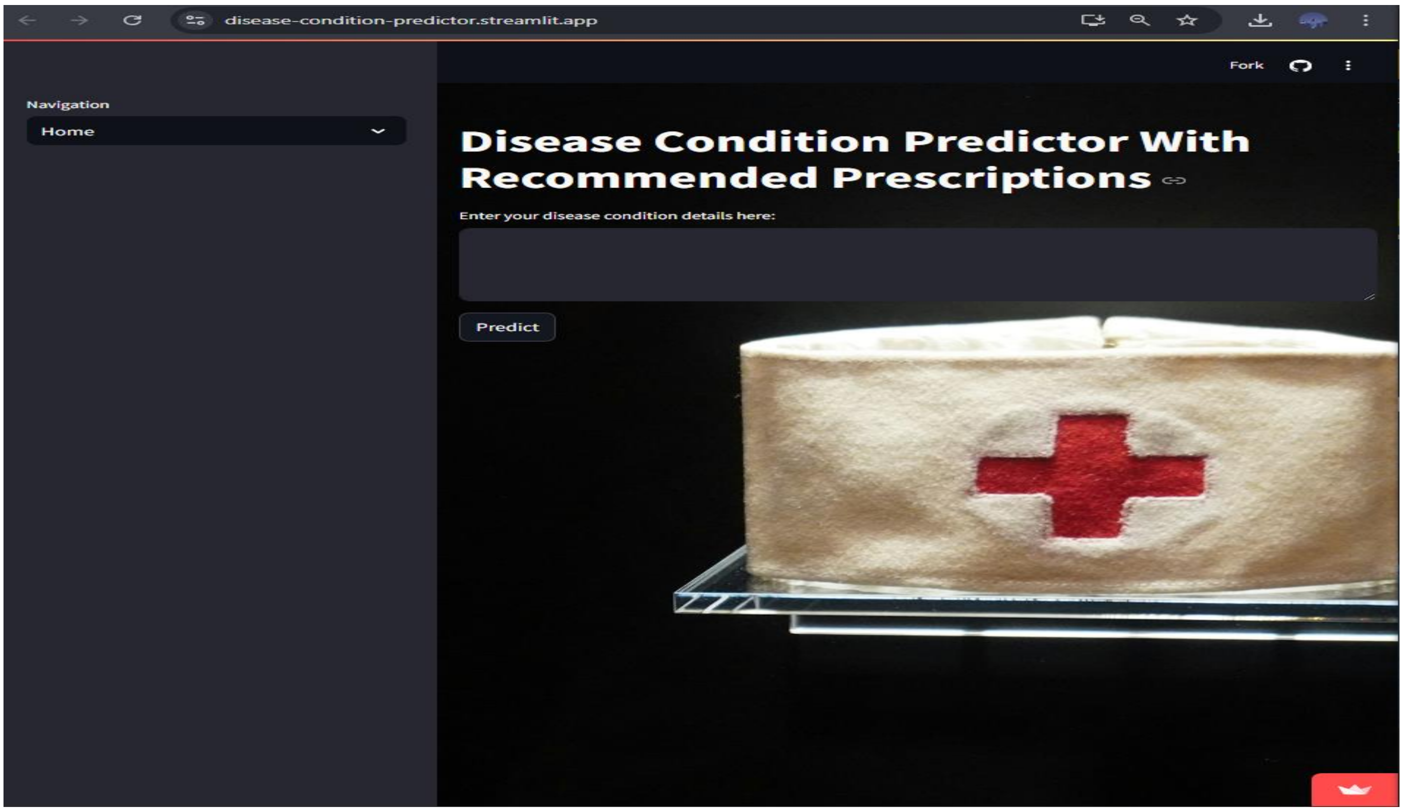
```python
st.title('Disease Condition Predictor With Recommended Prescriptions')

# Main Function

condition = st.text_area("Enter your disease condition details here:")

if st.button('Predict'):
    if condition == '':
        st.warning('No input Given')
    else:
        prediction,drugs = predictor(condition)
        st.error(f"Predicted Disease Condition: {prediction}")
        st.write(f"Top Recommended medications: ")
        st.success(", ".join(drugs))


#Drugs Recommender By Rating
def drugs(condition,rating):
    data = df[(df['condition']==condition) & (df['rating']==rating)]
    top = data.sort_values(by='rating', ascending=False).head(10)
    a = top[['drugname','rating']].reset_index(drop=True)
    return a


#Drugs recommender By usefullness & ratings
def drugstop(condition,rating,useful):
    data = df[(df['condition'] == condition) & (df['rating'] == rating)]
    top = data.groupby(['drugname', 'rating'])['usefulcount'].mean().reset_index(name='usefulcount').sort_values(by=['usefulcount','rating'],ascending=False).head(10)
    top = top[top['usefulcount']>useful]
    return top




#Sidebar

tool = st.sidebar.selectbox('Navigation',['Home','Drugs Recommender By Rating','Drugs recommender By usefullness & ratings'])




if tool == 'Drugs Recommender By Rating':

    st.title('Drugs Recommender By Rating')
    condition = st.selectbox('Whats your Disease Condition',df['condition'].unique())
    rating = st.slider('Rating',min_value=5,max_value=10)

    if st.button('Recommend'):
        recommendations = drugs(condition,rating)
        st.write(recommendations[['drugname','rating']],index=False)
        st.write("""_By Rasheed Farooqui._""")
```

6. Results

The results of using the Disease-Condition Predictor app were highly positive. Users could input their reviews and immediately receive a predicted disease condition based on the text they provided. The app proved to be accurate and responsive, leveraging the pre-trained Support Vector Classifier (SVC) model, which achieved a 90.26% accuracy rate during testing. The app's interface, built with Streamlit, was user-friendly, enabling seamless interaction. Doctors and healthcare professionals could utilize the app to gain insights from patient reviews, potentially aiding in more informed prescribing decisions and personalized healthcare recommendations.

Screen Shots

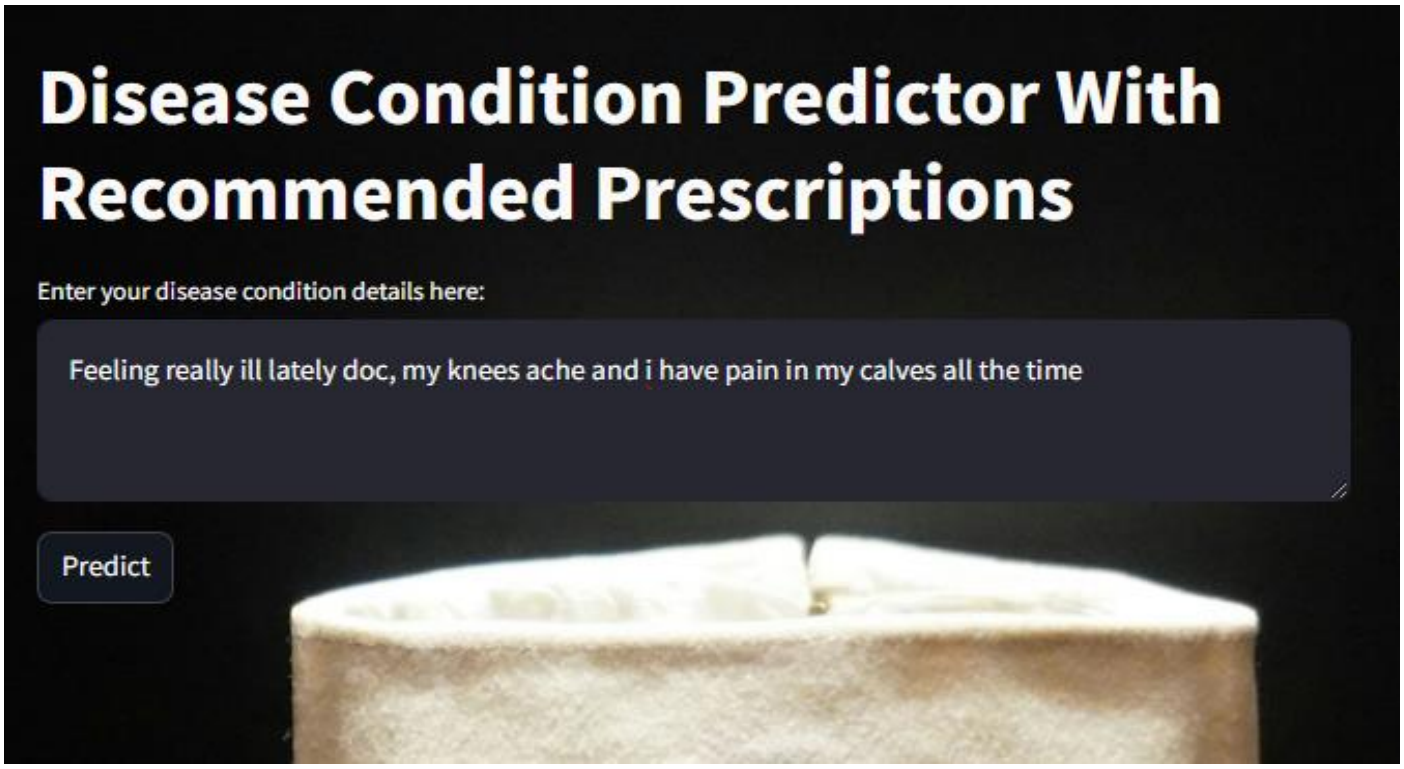
- Streamlit Interface



Disease Condition Predictor With Recommended Prescriptions

Enter your disease condition details here:

- Writing the condition

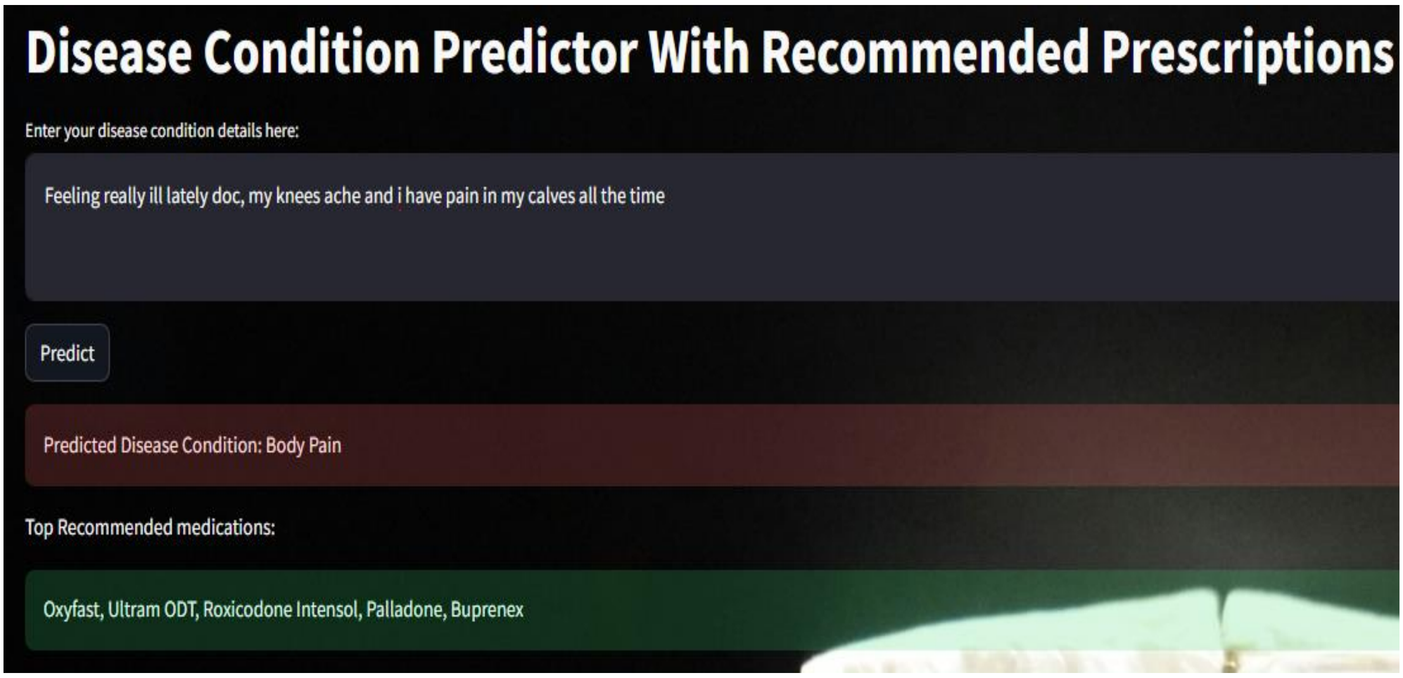


Disease Condition Predictor With Recommended Prescriptions

Enter your disease condition details here:

Feeling really ill lately doc, my knees ache and i have pain in my calves all the time

- Submitting the text by clicking predict



Disease Condition Predictor With Recommended Prescriptions

Enter your disease condition details here:

Feeling really ill lately doc, my knees ache and i have pain in my calves all the time

Predicted Disease Condition: Body Pain

Top Recommended medications:

Oxyfast, Ultram ODT, Roxicodone Intensol, Palladone, Buprenex

7. Conclusion and Future Enhancements

7.1 Conclusion

In conclusion, the Disease-Condition Predictor app successfully demonstrates the potential of combining natural language processing (NLP) and machine learning (ML) techniques to assist in healthcare. By analyzing patient reviews and accurately predicting their related conditions, the app serves as a valuable tool for doctors and healthcare professionals. It streamlines the process of understanding patient experiences and conditions, providing actionable insights that can enhance patient care and treatment outcomes. The app's accuracy, ease of use, and real-time performance suggest its applicability in clinical settings, paving the way for more personalized and data-driven healthcare solutions.

7.2 Future Enhancements

- **Database Integration with SQL and Python Flask Backend:** The current model can be enhanced by integrating a database to store and retrieve patient data, reviews, drug information, and conditions. By using SQL for database management and Python Flask as the backend, this integration would allow the application to handle larger datasets efficiently, ensure data persistence, and improve scalability. This will also enable storing users' input, managing patient information securely, and providing advanced querying capabilities.

- **Real-Time Prediction API:** Another enhancement would be developing a REST API using Flask that allows real-time prediction requests from external applications or mobile interfaces, enabling easier access for healthcare providers.

- **Improved Personalization:** Implement more advanced recommendation systems, using patient history or additional features like demographic data, to provide more accurate and tailored healthcare recommendations.

- **Enhanced Model Performance:** Continuously improve the accuracy and efficiency of the machine learning models by incorporating more complex algorithms like neural networks or by experimenting with hybrid models to better capture the nuances in the dataset.

- **User Authentication and Security:** Implementing user authentication features for doctors and healthcare providers could ensure that the data is accessed securely, safeguarding sensitive medical information.

- **Mobile App Integration:** Future plans could include developing a mobile app to make the tool accessible on smartphones, providing healthcare professionals with the ability to predict conditions on-the-go.