DRUG RECOMMENDATION USING SENTIMENT ANALYSIS

Mini Project Report

Submitted by

P SHRISHTI CLINT

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY-683577, ERNAKULAM(DIST)
FEBRUARY 2022

DECLARATION

I, **P SHRISHTI CLINT** hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of our original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date :

Place: Angamaly

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project report titled **”Drug Recommendation using Sentiment Analysis”** submitted by **P SHRISHTI CLINT** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by me during the year 2022.

Project Guide **Head of the Department**

Submitted for the viva-voice held on . . . . at . . . .

ACKNOWLEDGEMENT

I am deeply grateful to **Dr.Manoj George** , Principal, FISAT, Angamaly for providing me with adequate facilities, way and means by which I am able to complete my mini project work and I express my sincere thanks to **Dr. C. Sheela** ,Vice Principal FISAT, Angamaly.

My sincere thanks to **Dr.Deepa Mary Mathwes**, Head of the department of MCA, FISAT, who had been a source of inspiration. I express heartiest thanks to **Mr.Anju Prasanth** my project guide for encouragement and valuable suggestion .I express my heartiest gratitude to scrum master **Mr. Sujesh Lal** and the faculty members in our department for their constant encouragement and never ending support throughout the project.I would also like to express my sincere gratitude to the lab faculty members for their guidance

Finally I express my thanks to all my friends who gave me wealth of suggestion for successful completion of this project.

ABSTRACT

Since corona virus has shown up, inaccessibility of legitimate clinical resources is at its peak, like the shortage of specialists and healthcare workers, lack of proper equipment and medicines etc.

The entire medical fraternity is in distress, which results in numerous individual's demise. Due to unavailability, individuals started taking medication independently without appropriate consultation, making the health condition worse than usual.

As of late, machine learning has been valuable in numerous applications, and there is an increase in innovative work for automation. This paper intends to present a drug recommender system that can drastically reduce specialists heap.

We build a medicine recommendation system that uses patient reviews to predict the sentiment using various vectorization processes like Bow, TF-IDF,Word2Vec, and Manual Feature Analysis, which can help recommend the top drug for a given disease by different classification algorithms

Contents

1 Introduction 8

2 PROOF OF CONCEPT 9

2.1 Existing System 9

3 PROOF OF CONCEPT 10

3.1 Proposed System 10

4 IMPLEMENTATION 11

4.1 System Architecture 14

4.2 Dataset 14

4.3 Modules 14

4.3.1 SENTIMENT SEPARATION 14

4.3.2 PREPOSSESSING 14

4.3.3 SEGMENTATION 15

4.3.4 CLASSIFICATION 15

4.3.5 DEPLOYMENT 15

4.4 ALGORITHM 15

5 RESULT ANALYSIS 17

6 CONCLUSION AND FUTURE SCOPE 18

6.1 Conclusion 18

6.2 Future Scope 19

7 CODING 20

7.1 Training.ipynb 20

7.2 app.py 27

8 SCREEN SHOTS 29

9 REFERENCES 32

Chapter 1

Introduction

Clinical blunders are very regular nowadays. Over 200 thousand individuals in China and 100 thousand in the USA are affected every year because of prescription mistakes. Over 40since specialists compose the solution as referenced by their knowledge, which is very restricted .

Every day a new study comes up with accompanying more drugs, tests, accessible for clinical staff every day. Accordingly, it turns out to be progressively challenging for doctors to choose which treatment or medications to give to a patient based on indications, past clinical history.

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

Chapter 2

PROOF OF CONCEPT

2.1 Existing System

Covid-19 pandemic has rapidly affected our day to day life. Everyone are restricted to be within their house In such a situation, they are not getting an opportunity to have a direct contact the health department.
It is not possible to take medication without the help of individuals from medical, what patients can do is either visit medical centre or use telephone communcation.

Chapter 3

PROOF OF CONCEPT

3.1 Proposed System

This Project is based on sentiment analysis of the drug whether the drug should be given for patients, it is advisable or not to the patients. This project is implemented using Natural Language processing using a bag of words model and other techniques like vectorization to analyze the drug reviews.

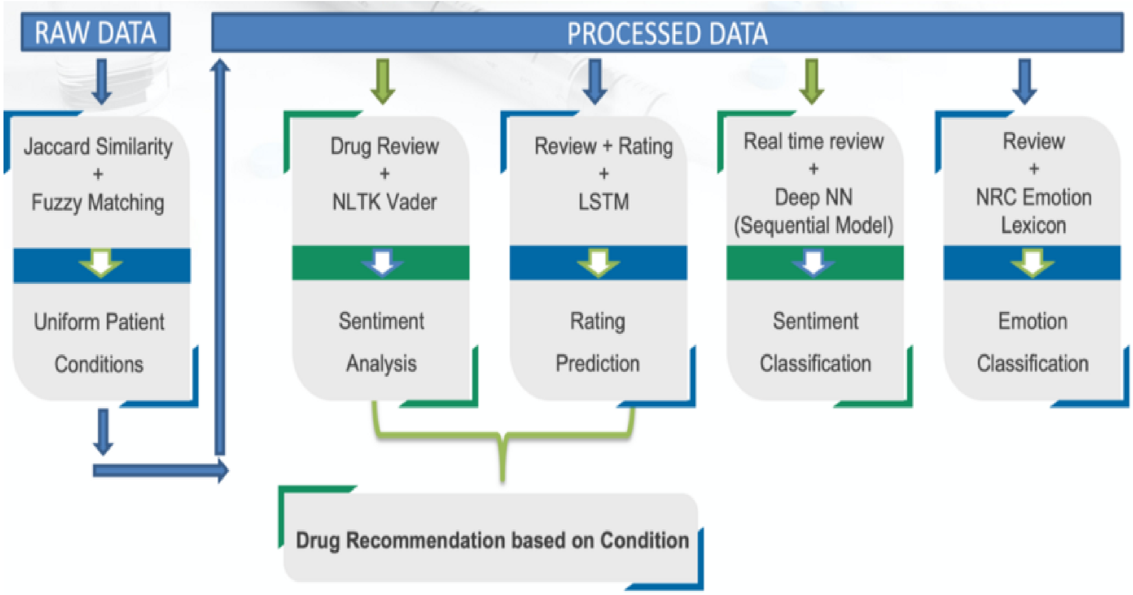
Chapter 4

IMPLEMENTATION

The condition and drug column were joined with review text because the condition and medication words also have predictive power. Before proceeding to the feature extraction part, it is critical to clean up the review text before vectorization. This process is also known as text preprocessing. We first cleaned the reviews after removing HTML tags, punctuations, quotes, URLs, etc. The cleaned reviews were lowercased to avoid duplication, and tokenization was performed for converting the texts into small pieces called tokens. Additionally, stopwords,for example, “a, to, all, we, with, etc.,”were removed from the corpus. The tokens were gotten back to their foundations by performing lemmatization on all tokens. For sentiment analysis, labeled every single review as positive and negative based on its user rating. If the user rating range between 6 to 10, then the review is positive else negative

After text preprocessing, a proper set up of the data required to build classifiers for sentiment analysis. Machine learning algorithms can’t work with text straightforwardly; it should be changed over into numerical format. In particular, vectors of numbers. A well known and straightforward strategy for feature extraction with text information used in this research is the bag of words (Bow), TF-IDF, Word2Vec.

Figure 4.1: Densenet121 architecture with the convolutional (112*112) layer at the beginning and five dense blocks.


RAW DATA
PROCESSED DATA
Jaccard Similarity + Fuzzy Matching
Uniform Patient Conditions
Drug Review + NLTK Vader
Sentiment Analysis
Review + Rating + LSTM
Rating Prediction
Real time review + Deep NN (Sequential Model)
Sentiment Classification
Review + NRC Emotion Lexicon
Emotion Classification
Drug Recommendation based on Condition

We used NLTK VADER (Valence Aware Dictionary and sentiment Reasoner) sentiment analyzer which is a lexicon and rule-based sentiment analysis tool. Since most of the drug reviews are written in informal language, we decided to use VADER as it works well on the non-technical, social media content and product reviews. In this approach, each word in the lexicon is rated as to whether it is positive, negative or neutral. The final sentiment value which is an aggregate score is called compound score. The compound score is a metric that calculates the sum of all lexicon ratings and normalizes between -1 to 1. As we can see from the below plot, sentiment polarity is quite uniformly distributed with a peak at 0, which implies that the majority of reviews were classified as neutral.

4.1 System Architecture

4.2 Dataset

The dataset used in this research is Drug Review Dataset (Drugs.com) taken from the UCI ML repository. This dataset contains six attributes, name of drug used (text), review (text) of a patient, condition (text) of a patient, useful count (numerical) which suggest the number of individuals who found the review helpful, date (date) of review entry, and a 10-star patient rating (numerical) determining overall patient contentment. It contains a total of 215063 instances from this dataset we train our models and choose one with maximum accuracy

4.3 Modules

4.3.1 SENTIMENT SEPARATION

For the classification task, data were divided into a train (80%), validation (10%), and test (10%) partitions

4.3.2 PREPOSSESSING

Text Pre-Processing using Snowball Stemmer:

1. Delete HTML 2. Removing special characters and keeping only letters 3. Convert to lower-case 4. Removing Stop words 5. Stemming 6. Joining stemming words

Text Pre-Processing using Spacy Tokenizer:

1. Creating the token object, which is used to create documents with linguistic annotations 2. Lemmatizing each token and converting each token into lowercase 3. Removing stop words 4. Return preprocessed list of tokens

4.3.3 SEGMENTATION

Sentiment Analysis:

Vader Sentiment Analysis was performed on the preprocessed reviews. Categorizes reviews into Positive(¿=0.05), Neutral(between -0.05 0.05) and Negative(¡=-0.05)

4.3.4 CLASSIFICATION

There is one classification tasks in this research, to distinguish whether a review is a good review or bad review.

4.3.5 DEPLOYMENT

Model deployment is simply the engineering task of exposing an ML model to real use. The term is often used quite synonymously with making a model available via real-time APIs.

4.4 ALGORITHM

Understanding Multinminal Naive Bayes (MNB)

Multinomial naive Bayes (MNB) assumes that all attributes (i.e., features) are independent of each other given the context of the class, and it ignores all dependencies among attributes. However, in many real-world applications, the attribute independence assumption required by MNB is often violated and thus harms its performance. To weaken this assumption, one of the most direct ways is to extend its structure to represent explicitly attribute dependencies by adding arcs between attributes. On the other hand, although a Bayesian network can represent arbitrary attribute dependencies, learning an optimal Bayesian network from high-dimensional text data is almost impossible. The main reason is that learning the optimal structure of a Bayesian network from high-dimensional text data is extremely time and space consuming. Thus, it would be desirable if a multinomial Bayesian network model can avoid structure learning and be able to represent attribute dependencies to some extent. In this paper, we propose a novel model called structure extended multinomial naive Bayes (SEMNB). SEMNB alleviates the attribute independence assumption by averaging all of the weighted one-dependence multinomial estimators. To learn SEMNB, we propose a simple but effective learning algorithm without structure searching. The experimental results on a large suite of benchmark text datasets show that SEMNB significantly outperforms MNB and is even markedly better than other three state-of-the-art improved algorithms including TDM, DWMNB, and Rw, cMNB.:

Chapter 5

RESULT ANALYSIS

The result of the proposed project Drug Recommendation Using Sentiment Analysis lies in developing a handy web app that can successfully predict whether a review given by the user is positive or negative.

The proposed system takes user review as input and outputs whether a drug can be used by the patients.

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

We aimed to extract effective inferences from our data that would benefit drug users, pharma companies and clinicians by receiving feedback of the drug based on opinion mining. We recommended top drugs for a given condition based on the sentiment score using VADER and LSTM model rating prediction. We also analyzed the emotion inclination towards a drug using 8 emotions. We get the best predictions with MLP + TF-IDF model, with an accuracy of 83 percentage outperforming baseline models. We trained our predictive models using NLP bag-of-words models (TF-IDF, Hashing) along with different tokenizers as part of text pre-processing. We also utilized Facebook's fastText to learn word embeddings and observed similarity among word groupings using t-SNE. Lastly, one of the most important features of our project is our interactive web application accomplishing two main goals, showcasing useful data insights and achieving real-time classification of sentiment.

6.2 Future Scope

This Drug Recommender application developed based on a normalized recommended score, will help the doctors to view the top rated drugs for a particular condition/disease by using a user-friendly and interactive web application.

Although we tried removing spam reviews while preprocessing the data, the dataset had too many duplicate reviews. A proper spam removal model could be developed for the same in the future.We tried implementing a recommendation system using content based and collaborative item-based filtering. However, the huge data set led to sparse matrix and inaccurate results. A hybrid system could be used in the future. (The item based Collaborative Filtering code is uploaded)

.

Chapter 7

CODING

7.1 Training.ipynb

1. **Mount Drive**

```
from google.colab import drive
drive.mount(’/content/drive’)
```

2. **Importing necessary libraries**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import spacy
import string
from nltk.corpus import stopwords
from collections import Counter
from nltk.stem.porter import PorterStemmer
import warnings
warnings.filterwarnings(”ignore”)
print(’Library Importing Complete’)
```

3. **Loading data**

```
!git clone https://github.com/Dennis-Neduvelil/Covid-pneumonia-DataSet.git
```

4. **Split data into Training / Validation / Testing set**

```
from sklearn.model selection import train test split
class DataFrame Preprocessor():

def init(self):

print(”Preprocessor object created”)

def preprocess(self,data):
data[’rating’] = pd.to-numeric(data[’rating’],errors=’coerce’)
data[’Sentiment’] = np.where(data[’rating’] ¿ 6, 1, 0)
data= data[[’review’,’Sentiment’]]
x = data[’review’]
y = data[’Sentiment’]
return train-test-split(x,y,test-size=0.1, random-state=0)
```

Data Preprocessing

```
class DataFrame-Preprocessor():

def –init–(self,n-rare-words):
self.n-rare-words = 10
print(”Preprocessor object created”)
def remove punctuation(self,text):
PUNCT TO REMOVE = string.punctuation
”””custom function to remove the punctuation”””
return text.translate(str.maketrans(”, ”, PUNCT-TO-REMOVE))
def remove stopwords(self,text):

STOPWORDS = set(stopwords.words(’english’))
”””custom function to remove the stopwords”””
return ” ”.join([word for word in str(text).split() if word not in STOPWORDS])
def Get Most Commom(self,data):

cnt = Counter()
for text in df[”review”].values:
for word in text.split():
cnt[word] += 1
return cnt.most-common(10)

def remove freqwords(self,text):
FREQWORDS = set([w for (w, wc) in count])
”””custom function to remove the frequent words”””
return ” ”.join([word for word in str(text).split() if word not in FREQWORDS])
def remove rarewords(self,text):
```

```
RAREWORDS = set([w for (w, wc) in count[:-self.n-rare-words-1:-1]])
”””custom function to remove the rare words”””
return ” ”.join([word for word in str(text).split() if word not in RAREWORDS])

def stem-words(self,text):
stemmer = PorterStemmer()
return ” ”.join([stemmer.stem(word) for word in text.split()])

def Text Preprocessing(self,data):

try:

data = data[[’review’,’rating’]]
data[”review”] = data[”review”].apply(lambda text:
self.remove-punctuation(text))
data[”review”] = data[”review”].apply(lambda text: self.remove-stopwords(text))
data[”review”] = data[”review”].apply(lambda text: self.remove-freqwords(text))
data[”review”] = data[”review”].apply(lambda text: self.remove-rarewords(text))
data[”review”] = data[”review”].apply(lambda text: self.stem-words(text))
data = data.astype(str).apply(lambda x: x.str.encode(’ascii’,
’ignore’).str.decode(’ascii’))
data[’review’] = data[’review’].str.replace(’+̣’, ”)
return data
except ValueError as ve:
raise(ValueError(”Error in Text Preprocessing ”.format(ve)))
```

5. **Creating Models**

```
from tensorflow import keras
```

```
class RNN-Bidirectional-lstm-Build-Pack():

def -init-(self,
input-length,
output-length,
vocab-size,
optimizer,
loss,
metrics,
batch-size,
epochs,
verbose):


self.input-length =200
self.output-length= 200
self.vocab-size = 33068
self.optimizer = ’adam’
self.loss = ’binary-crossentropy’
self.metrics = [’acc’]
self.batch-size = 256
self.epochs = 20
self.verbose = 1


print(”Tokenizer object created”)

def build-rnn(self,vocab-size,output-dim, input-dim):

model = Sequential([
keras.layers.Embedding(self.vocab-size,output-dim = self.output-length,
input-length = self.input-length),
keras.layers.BatchNormalization(),
keras.layers.Bidirectional(keras.layers.LSTM(256,return-sequences=True)),
keras.layers.GlobalMaxPool1D(),
keras.layers.Dense(225,activation=’relu’),
keras.layers.Dropout(0.3),
keras.layers.Dense(150,activation=’relu’),
keras.layers.Dropout(0.2),
keras.layers.Dense(95,activation=’relu’),
keras.layers.Dropout(0.2),
keras.layers.Dense(64,activation=’relu’),
keras.layers.Dropout(0.1),
keras.layers.Dense(34,activation=’relu’),
keras.layers.Dropout(0.1),
keras.layers.Dense(32,activation=’relu’),
keras.layers.Dense(output-dim, activation=’sigmoid’)
])


return model

def Compile-and-Fit(self,rnn-model):
try:
rnn-model.compile(optimizer=self.optimizer, loss=self.loss, metrics=self.metrics)

rnn-model.fit(x-pad-train,
y-train,
batch-size=self.batch-size,
epochs=self.epochs,
verbose= self.verbose)


score = rnn-model.evaluate(x-pad-valid, y-test, verbose=1)



print(”Loss:


return rnn-model
except ValueError as Model-Error:
raise(ValueError(”Model Compiling Error ”.format(Model-Error)))
```

7.2 app.py

```
from flask import Flask,render-template,url-for,request
import numpy as np
import pickle
import pandas as pd
from keras.models import load-model
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad-sequences
from keras.models import model-from-json
from numpy import array


app=Flask(name)
model = load-model(”rnn-model.h5”)
with open(’tokenizer.pickle’, ’rb’) as handle:
tokenizer = pickle.load(handle)


@app.route(’/’)

def home():
return render-template(’home.html’)



@app.route(’/predict’,methods=[’POST’])
def predict():
max-length = 200
if request.method == ’POST’:
```

```
review = request.form[’review’]
data = [review]
tokenizer.fit-on-texts(data)
enc = tokenizer.texts-to-sequences(data)
enc=pad-sequences(enc, maxlen=max-length, padding=’post’)
my-prediction = model.predict(array([enc][0]))[0][0]
class1 = model.predict-classes(array([enc][0]))[0][0]
return render-template(’result.html’,prediction = class1)


if name == ’main’:
app.run(debug=True)
```

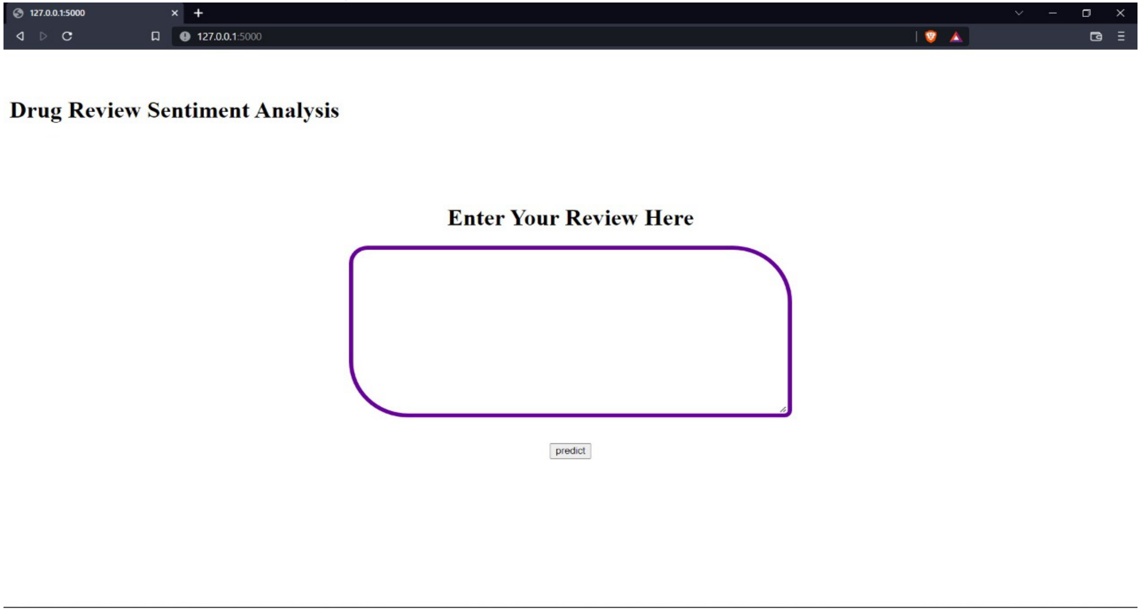
Chapter 8

SCREEN SHOTS

Here I add some sample screenshots of the proposed system which includes,

- Enter Review Screen
- Positive Prediction Screen
- Negative Prediction Screen

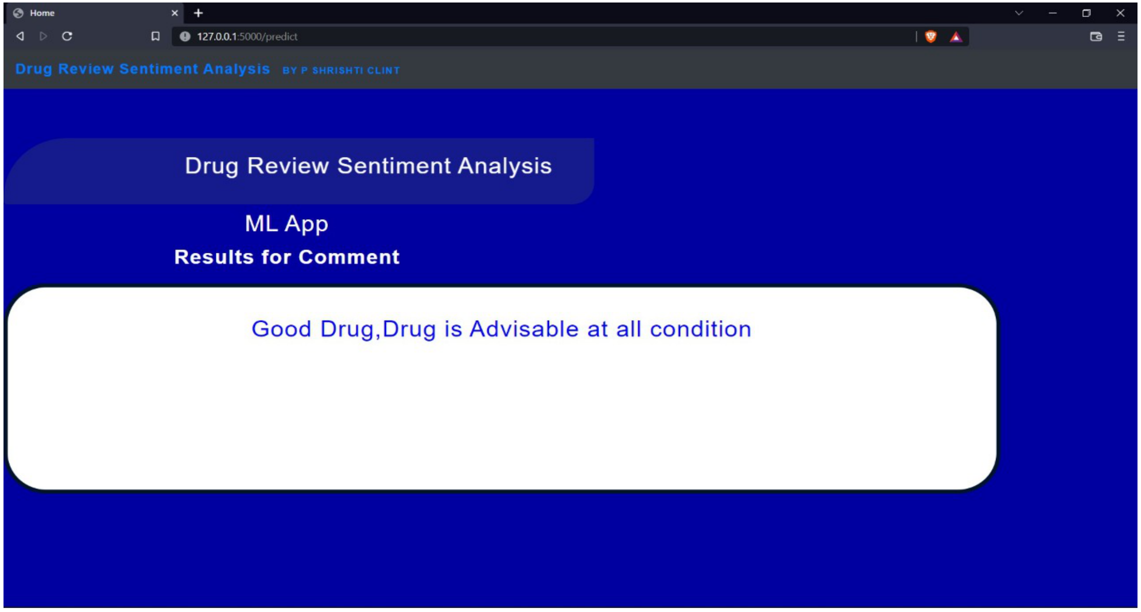
Figure 8.1: Enter Review Screen



Drug Review Sentiment Analysis

Enter Your Review Here

Figure 8.2: Positive Prediction Screen



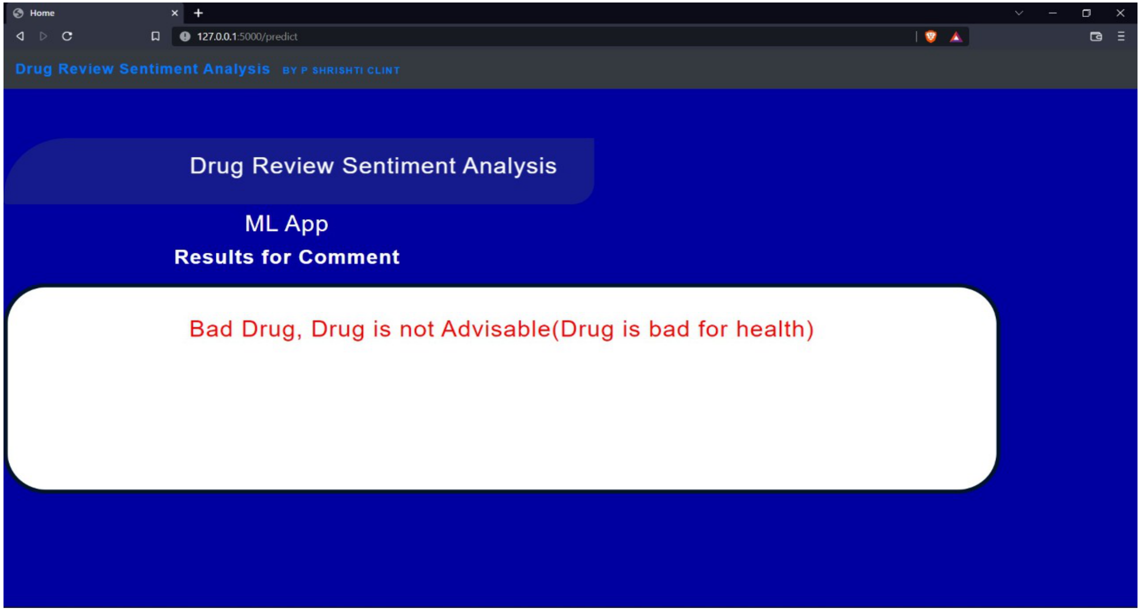
Drug Review Sentiment Analysis

ML App

Results for Comment

Good Drug,Drug is Advisable at all condition

Figure 8.3: Negative Prediction Screen



Drug Review Sentiment Analysis

ML App

Results for Comment

Bad Drug, Drug is not Advisable(Drug is bad for health)

Chapter 9

REFERENCES

1. www.google.com

2. www.youtube.com

3. www.wikipedia.com

4. Telemedicine, https://www.mohfw.gov.in/pdf/Telemedicine.pdf

5. CM, Burkle CM, Lanier WL. Medication errors: an overview for clinicians. Mayo Clin Proc. 2014 Aug;89(8):1116-25. CHEN, M. R., WANG, H. F. (2013). The reason and prevention of hospital medication errors. Practical Journal of Clinical Medicine, 4.

6. Drug Review Dataset, https://archive.ics.uci.edu/ml/datasets/Drug2BReview

7. Fox, Susannah, and Maeve Duggan. ”Health online 2013. 2013.” URL: http://pewinternet.org/Reports/2013/Health-online.aspx

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

9. Fox, Susannah Duggan, Maeve. (2012). Health Online 2013. Pew Research Internet Project Report.

10. T. N. Tekade and M. Emmanuel, ”Probabilistic aspect mining approach for interpretation and evaluation of drug reviews,” 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, 2016, pp. 1471-1476, doi: 10.1109/SCOPES.2016.7955684.

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

12. Leilei Sun, Chuanren Liu, Chonghui Guo, Hui Xiong, and Yanming Xie. 2016. Data-driven Automatic Treatment Regimen Development and Recommendation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16). Association for Computing Machinery, New York, NY, USA, 1865–1874. DOI:https://doi.org/10.1145/293

13. V. Goel, A. K. Gupta and N. Kumar, ”Sentiment Analysis of Multilingual Twitter Data using Natural Language Processing,” 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2018, pp. 208-212, doi: 10.1109/CSNT.2018.8820254.

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