SOURCE CODE

```
from tkinter import messagebox

from tkinter import *

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

from tkinter.filedialog import askopenfilename

from CustomButton import TkinterCustomButton

from sklearn.model_selection import train_test_split

from sklearn.metrics import precision_score

from sklearn.metrics import recall_score

from sklearn.metrics import f1_score

from sklearn.metrics import accuracy_score

import pandas as pd

import numpy as np

import pickle

import os


from string import punctuation

from nltk.corpus import stopwords

import nltk

from nltk.stem import WordNetLemmatizer
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.neural_network import MLPClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.linear_model import RidgeClassifier
from numpy import dot
from numpy.linalg import norm

main = Tk()
main.title("Integrated ML with NLP framework for Drug Recommendation System")
main.geometry("1300x1200")

global filename
global dataset
global X, Y
global X_train, X_test, y_train, y_test
accuracy = []
precision = []
recall = []
fscore = []

stop_words = set(stopwords.words('english'))
```

```
lemmatizer = WordNetLemmatizer()


global drug_name, condition, review, rating
global tfidf_vectorizer
global classifier


def cleanPost(doc):
    tokens = doc.split()
    table = str.maketrans('', '', punctuation)
    tokens = [w.translate(table) for w in tokens]
    tokens = [word for word in tokens if word.isalpha()]
    tokens = [w for w in tokens if not w in stop_words]
    tokens = [word for word in tokens if len(word) > 1]
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    tokens = ' '.join(tokens)
    return tokens


def uploadDataset():
    global filename
    global dataset
    text.delete('1.0', END)
    filename = askopenfilename(initialdir = "Dataset")
    tf1.insert(END,str(filename))
    text.insert(END,"Dataset Loaded\n\n")
```

```
    dataset = pd.read_csv(filename,sep="\t",nrows=5000)
    text.insert(END,str(dataset.head()))
    label = dataset.groupby('rating').size()
    label.plot(kind="bar")
    plt.title("Ratings Graph")
    plt.show()



def preprocessDataset():
    global X, Y
    global X_train, X_test, y_train, y_test
    global drug_name, condition, review, rating
    global dataset
    text.delete('1.0', END)
    if os.path.exists('model/data.npy'):
        data = np.load("model/data.npy")
        drug_name = data[0]
        condition = data[1]
        review = data[2]
        rating = data[3]
    else:
        for i in range(len(dataset)):
            dname = dataset.get_value(i,"drugName")
            cond = dataset.get_value(i,"condition")
```

```
            reviewText = dataset.get_value(i,"review")
            ratings = dataset.get_value(i,"rating")
            reviewText = str(reviewText)
            reviewText = reviewText.strip().lower()
            reviewText = cleanPost(reviewText)
            drug_name.append(dname)
            condition.append(cond)
            review.append(reviewText)
            rating.append(ratings-1)
            print(i)
        data = [drug_name,condition,review,rating]
        data = np.asarray(data)
        np.save("model/data",data)
    text.insert(END,"Reviews after cleaning and preprocessing\n\n")
    text.insert(END,str(review))
    label = dataset.groupby('drugName').size().head(20)
    label.plot(kind="bar")
    plt.title("Top 20 Drug Name Graph")
    plt.show()

def TFIDFExtraction():
    global drug_name, condition, review, rating
    global tfidf_vectorizer
    text.delete('1.0', END)
```

```
    global X, Y
    global X_train, X_test, y_train, y_test
    tfidf_vectorizer = TfidfVectorizer(stop_words=stop_words, use_idf=True,
smooth_idf=False, norm=None, decode_error='replace', max_features=700)
    tfidf = tfidf_vectorizer.fit_transform(review).toarray()
    df = pd.DataFrame(tfidf, columns=tfidf_vectorizer.get_feature_names())
    text.insert(END,str(df)+"\n\n")
    text.insert(END,str(df.values[0]))
    df = df.values
    X = df[:, 0:df.shape[1]]
    Y = rating
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)



def test(cls,name):
    predict = cls.predict(X_test)
    acc = accuracy_score(y_test,predict)*100
    p = precision_score(y_test,predict,average='macro') * 100
    r = recall_score(y_test,predict,average='macro') * 100
    f = f1_score(y_test,predict,average='macro') * 100
    text.insert(END,name+" Precision  : "+str(p)+"\n")
```

```
    text.insert(END,name+" Recall     : "+str(r)+"\n")
    text.insert(END,name+" F1-Score   : "+str(f)+"\n")
    text.insert(END,name+" Accuracy   : "+str(acc)+"\n\n")
    precision.append(p)
    accuracy.append(acc)
    recall.append(r)
    fscore.append(f)


def TrainML():
    global X, Y
    global X_train, X_test, y_train, y_test
    global classifier
    text.delete('1.0', END)
    if os.path.exists('model/lr.txt'):
        with open('model/lr.txt', 'rb') as file:
            lr_cls = pickle.load(file)
        file.close()
        test(lr_cls,"Logistic Regression")
    else:
        lr_cls = LogisticRegression(max_iter=500)
        lr_cls.fit(X,Y)
        test(lr_cls,"Logistic Regression")
        with open('model/lr.txt', 'wb') as file:
            pickle.dump(lr_cls, file)
```

```
    file.close()


if os.path.exists('model/svc.txt'):
    with open('model/svc.txt', 'rb') as file:
        svc_cls = pickle.load(file)
    file.close()
    test(svc_cls,"Linear SVC")
else:
    svc_cls = LinearSVC()
    svc_cls.fit(X,Y)
    test(svc_cls,"Linear SVC")
    with open('model/svc.txt', 'wb') as file:
        pickle.dump(svc_cls, file)
    file.close()


if os.path.exists('model/ridge.txt'):
    with open('model/ridge.txt', 'rb') as file:
        ridge_cls = pickle.load(file)
    file.close()
    test(ridge_cls,"Ridge Classifier")
else:
    ridge_cls = RidgeClassifier()
    ridge_cls.fit(X,Y)
    test(ridge_cls,"Ridge Classifier")
```

```
    with open('model/ridge.txt', 'wb') as file:
        pickle.dump(ridge_cls, file)
    file.close()

if os.path.exists('model/nb.txt'):
    with open('model/nb.txt', 'rb') as file:
        nb_cls = pickle.load(file)
    file.close()
    test(nb_cls,"Multinomial Naive Bayes")
else:
    nb_cls = MultinomialNB()
    nb_cls.fit(X,Y)
    test(nb_cls,"Multinomial Naive Bayes")
    with open('model/nb.txt', 'wb') as file:
        pickle.dump(nb_cls, file)
    file.close()

if os.path.exists('model/sgd.txt'):
    with open('model/sgd.txt', 'rb') as file:
        sgd_cls = pickle.load(file)
    file.close()
    test(sgd_cls,"SGDClassifier")
else:
    sgd_cls = MultinomialNB()
```

```
        sgd_cls.fit(X,Y)
        test(sgd_cls,"SGDClassifier")
        with open('model/sgd.txt', 'wb') as file:
            pickle.dump(sgd_cls, file)
        file.close()

    if os.path.exists('model/mlp.txt'):
        with open('model/mlp.txt', 'rb') as file:
            mlp_cls = pickle.load(file)
        file.close()
        test(mlp_cls,"Multilayer Perceptron Classifier")
        classifier = mlp_cls
    else:
        mlp_cls = MLPClassifier()
        mlp_cls.fit(X,Y)
        test(mlp_cls,"Multilayer Perceptron Classifier")
        with open('model/mlp.txt', 'wb') as file:
            pickle.dump(mlp_cls, file)
        file.close()
        classifier = mlp_cls


def graph():
```

```
    df = pd.DataFrame([['Logistic Regression','Accuracy',accuracy[0]],['Logistic
Regression','Precision',precision[0]],['Logistic Regression','Recall',recall[0]],['Logistic
Regression','FScore',fscore[0]],

                ['Linear SVC','Accuracy',accuracy[1]],['Linear
SVC','Precision',precision[1]],['Linear SVC','Recall',recall[1]],['Linear
SVC','FScore',fscore[1]],

                ['Ridge Classifier','Accuracy',accuracy[2]],['Ridge
Classifier','Precision',precision[2]],['Ridge Classifier','Recall',recall[2]],['Ridge
Classifier','FScore',fscore[2]],


['MultinomialNB','Accuracy',accuracy[3]],['MultinomialNB','Precision',precision[3]],['Multin
omialNB','Recall',recall[3]],['MultinomialNB','FScore',fscore[3]],


['SGDClassifier','Accuracy',accuracy[4]],['SGDClassifier','Precision',precision[4]],['SGDClas
sifier','Recall',recall[4]],['SGDClassifier','FScore',fscore[4]],

                ['MLP Classifier','Accuracy',accuracy[5]],['MLP
Classifier','Precision',precision[5]],['MLP Classifier','Recall',recall[5]],['MLP
Classifier','FScore',fscore[5]],


                ],columns=['Parameters','Algorithms','Value'])
    df.pivot("Parameters", "Algorithms", "Value").plot(kind='bar')
    plt.show()


def recommendDrug():
    text.delete('1.0', END)
    global X
    global drug_name, condition, review, rating
    global classifier
```

```
    global tfidf_vectorizer
    filename = askopenfilename(initialdir = "Dataset")
    testData = pd.read_csv(filename)
    testData = testData.values
    for i in range(len(testData)):
        review = cleanPost(testData[i,0].strip().lower())
        array = tfidf_vectorizer.transform([review]).toarray()
        predict = classifier.predict(array)[0]
        maxValue = 0
        dname = "none"
        print(str(array[0].shape)+" "+str(X.shape))
        for j in range(len(X)):
            score = dot(X[j], array[0])/(norm(X[j])*norm(array[0]))
            if score > maxValue:
                maxValue = score
                dname = drug_name[j]
        text.insert(END,"Disease Name: "+str(testData[i,0])+"\n")
        text.insert(END,"Recommended Drug: "+str(dname)+"\n")
        text.insert(END,"Predicted Ratings: "+str(predict)+"\n\n")



font = ('times', 15, 'bold')
title = Label(main, text='Integrated ML with NLP framework for Drug Recommendation System')
title.config(bg='HotPink4', fg='yellow2')
```

```
title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)


font1 = ('times', 13, 'bold')

ff = ('times', 12, 'bold')


l1 = Label(main, text='Dataset Location:')

l1.config(font=font1)

l1.place(x=50,y=100)


tf1 = Entry(main,width=60)

tf1.config(font=font1)

tf1.place(x=230,y=100)


uploadButton = TkinterCustomButton(text="Upload Drug Review Dataset", width=300,
corner_radius=5, command=uploadDataset)

uploadButton.place(x=50,y=150)


preprocessButton = TkinterCustomButton(text="Read & Preprocess", width=300,
corner_radius=5, command=preprocessDataset)

preprocessButton.place(x=400,y=150)


tfidfButton = TkinterCustomButton(text="TF-IDF Features Extraction", width=300,
corner_radius=5, command=TFIDFExtraction)
```

```
tfidfButton.place(x=790,y=150)

trainMLButton = TkinterCustomButton(text="Train ML models", width=300,
corner_radius=5, command=TrainML)
trainMLButton.place(x=50,y=200)

graphButton = TkinterCustomButton(text="Performance Comparison", width=300,
corner_radius=5, command=graph)
graphButton.place(x=400,y=200)

predictButton = TkinterCustomButton(text="Recommend Drug from Test Data", width=300,
corner_radius=5, command=recommendDrug)
predictButton.place(x=790,y=200)

font1 = ('times', 13, 'bold')
text=Text(main,height=20,width=130)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=250)
text.config(font=font1)

main.config(bg='plum2')
main.mainloop()
```

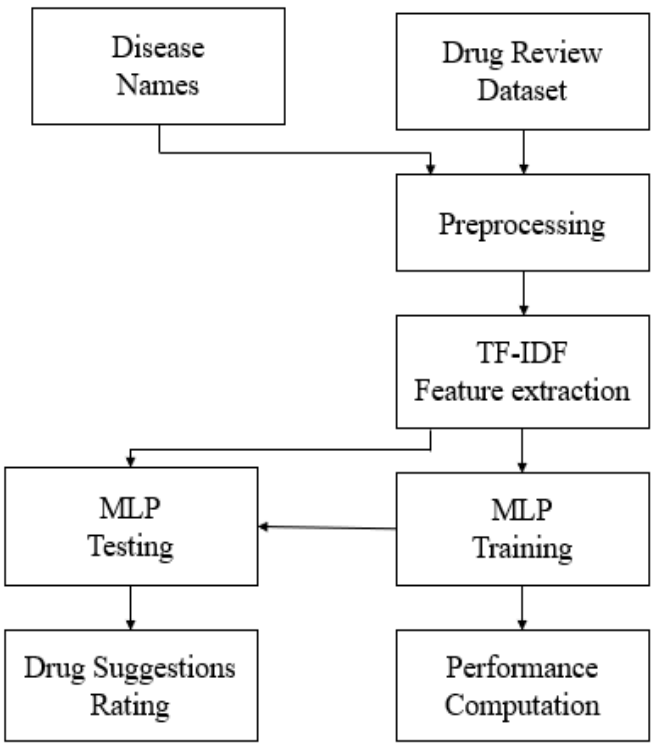
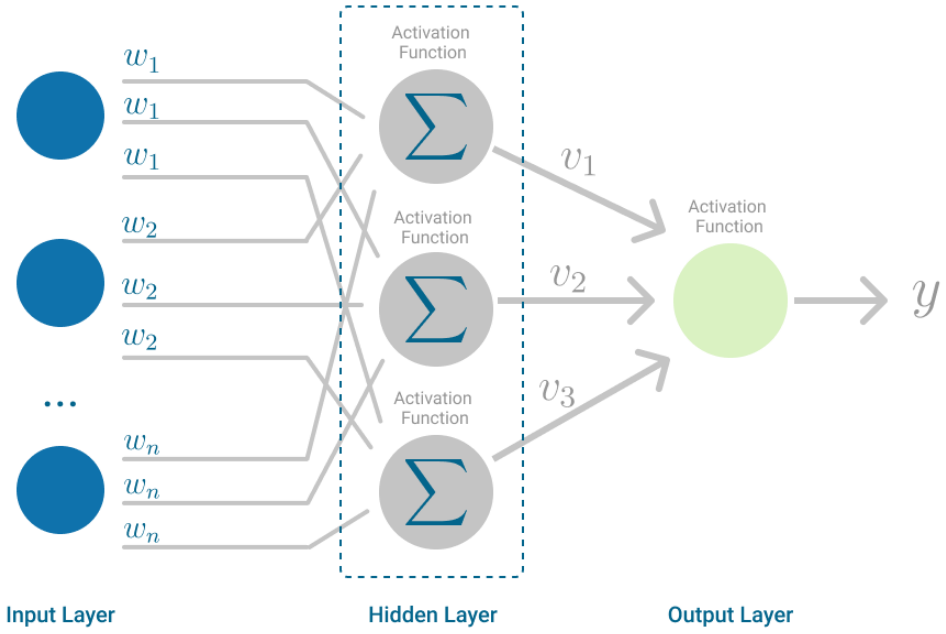


Fig. 1: Block Diagram of Proposed System.



Architecture of MLP.

RESULTS AND DISCUSSION

DRUGREVIEW Dataset

Data Set Information

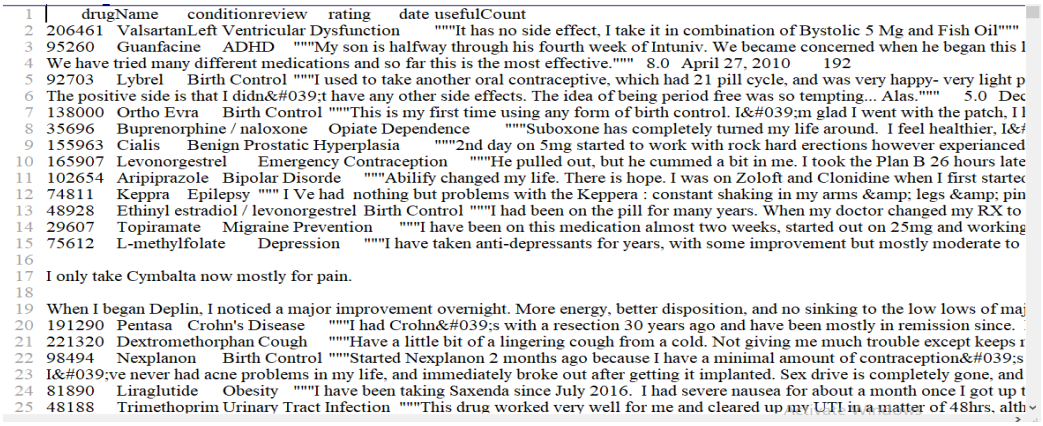
The dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting overall patient satisfaction. The data was obtained by crawling online pharmaceutical review sites. The intention was to study.

1. Sentiment analysis of drug experience over multiple facets, i.e., sentiments learned on specific aspects such as effectiveness and side effects,
2. The transferability of models among domains, i.e., conditions, and
3. The transferability of models among different data sources (see 'Drug Review Dataset (Druglib.com)').

The data is split into a train (75%) and a test (25%) partition (see publication) and stored in two.tsv (tab-separated-values) files, respectively.

Attribute Information

- drugName (categorical): name of drug.
- condition (categorical): name of condition.
- review (text): patient review.
- rating (numerical): 10-star patient rating.
- date (date): date of review entry.
- usefulCount (numerical): number of users who found review useful.



```
1    drugName    conditionreview    rating    date usefulCount
2  206461  ValsartanLeft Ventricular Dysfunction    """It has no side effect, I take it in combination of Bystolic 5 Mg and Fish Oil"""
3  95260   Guanfacine  ADHD    """My son is halfway through his fourth week of Intuniv. We became concerned when he began this l
4  We have tried many different medications and so far this is the most effective."""  8.0  April 27, 2010    192
5  92703   Lybrel  Birth Control   """I used to take another oral contraceptive, which had 21 pill cycle, and was very happy- very light p
6  The positive side is that I didn&#039;t have any other side effects. The idea of being period free was so tempting... Alas."""    5.0  Dec
7  138000  Ortho Evra  Birth Control   """This is my first time using any form of birth control. I&#039;m glad I went with the patch, I l
8  35696   Buprenorphine / naloxone  Opiate Dependence    """Suboxone has completely turned my life around.  I feel healthier, I&#
9  155963  Cialis  Benign Prostatic Hyperplasia    """2nd day on 5mg started to work with rock hard erections however experianced
10 165907  Levonorgestrel  Emergency Contraception    """He pulled out, but he cummed a bit in me. I took the Plan B 26 hours late
11 102654  Aripiprazole  Bipolar Disorde    """Abilify changed my life. There is hope. I was on Zoloft and Clonidine when I first startec
12 74811   Keppra  Epilepsy    """ I Ve had  nothing but problems with the Keppera : constant shaking in my arms &amp;amp; legs &amp;amp; pin
13 48928   Ethinyl estradiol / levonorgestrel  Birth Control   """I had been on the pill for many years. When my doctor changed my RX to
14 29607   Topiramate  Migraine Prevention    """I have been on this medication almost two weeks, started out on 25mg and working
15 75612   L-methylfolate  Depression    """I have taken anti-depressants for years, with some improvement but mostly moderate to
16
17 I only take Cymbalta now mostly for pain.
18
19 When I began Deplin, I noticed a major improvement overnight. More energy, better disposition, and no sinking to the low lows of maj
20 191290  Pentasa  Crohn's Disease    """I had Crohn&#039;s with a resection 30 years ago and have been mostly in remission since.
21 221320  Dextromethorphan Cough    """Have a little bit of a lingering cough from a cold. Not giving me much trouble except keeps r
22 98494   Nexplanon  Birth Control   """Started Nexplanon 2 months ago because I have a minimal amount of contraception&#039;s
23 I&#039;ve never had acne problems in my life, and immediately broke out after getting it implanted. Sex drive is completely gone, and
24 81890   Liraglutide  Obesity    """I have been taking Saxenda since July 2016.  I had severe nausea for about a month once I got up t
25 48188   Trimethoprim Urinary Tract Infection    """This drug worked very well for me and cleared up my UTI in a matter of 48hrs, alth
```

Figure 1. Sample dataset

In above screen first row represents dataset column names such as drug name, condition, review and rating and remaining rows contains dataset values and we will used above REVIEWS and RATINGS to trained machine learning models. Below is the test data which contains only disease name and machine learning will predict Drug name and ratings.

```
Disease_name
Rheumatoid Arthritis
Panic Disorde
Depression
Underactive Thyroid
Constipation
Urinary Tract Infection
High Blood Pressure
```

Figure 2. Dataset properties

In above test data we have only disease name and machine learning will predict ratings and drug names.

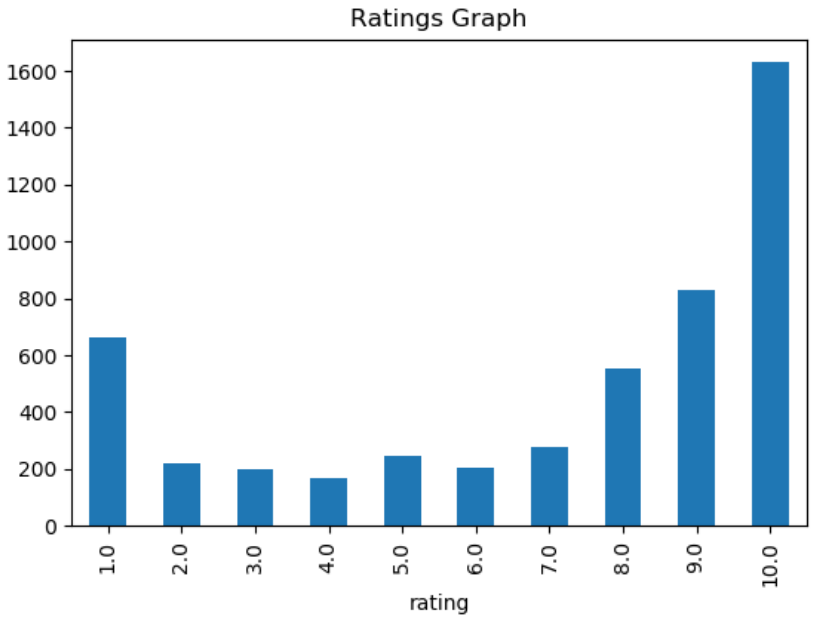


Figure 3. Drugs ratings graph

In above graph we can see dataset loaded and in graph x-axis represents ratings and y-axis represents total number of records which got that rating. Now close above graph and then click on 'Read & Pre-process Dataset' button to read all dataset values and then pre-process to remove stop words and special symbols and then form a features array.

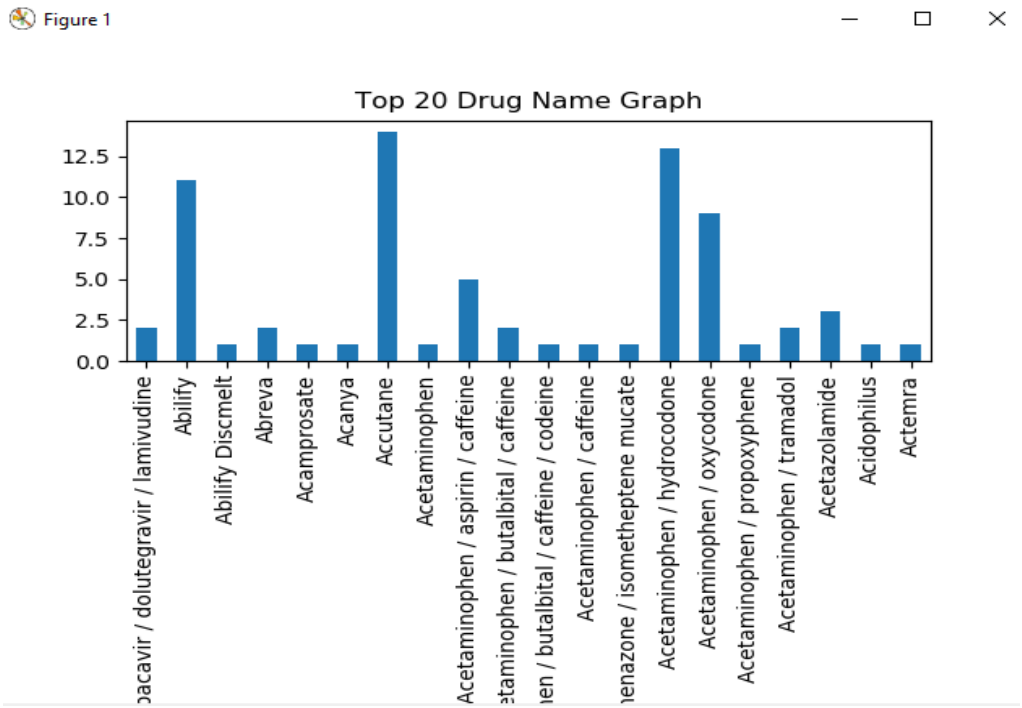


Figure 5. Drug names dataset

In above screen we can see from all reviews stop words and special symbols are removed and in graph I am displaying TOP 20 medicines exist in dataset. In above graph x-axis represents drug name and y-axis represents its count.

Table 1. Performance comparison

Method	Precision	Recall	F1-Score	Accuracy
Existing Logistic regression	80.54	79.30	79.27	76
Existing SVC	70.51	71.18	70,46	67.80
Existing Ridge classifier	66.786	37.72	42.78	55.1
Existing Multimodal navie bayes	41.32	47.98	43.14	47.19
Existing SGDC	41.324	47.18	43.44	47.49
Proposed MLP	99.96	99.72	99.84	99.9

In above table for each algorithm we calculate accuracy, precision, recall and FSCORE and in all algorithms MLP has got high performance.

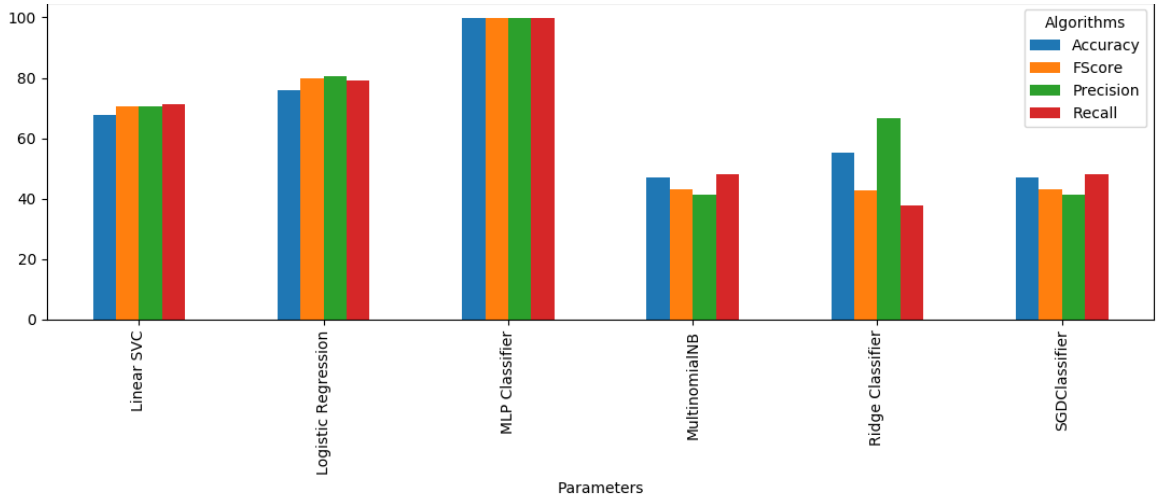


Figure 6. Performance comparison graph

In above graph x-axis represents algorithm name and y-axis represents accuracy, precision recall and FSCORE where each different colour bar will represent one metric and in above graph we can see MLP got high performance.

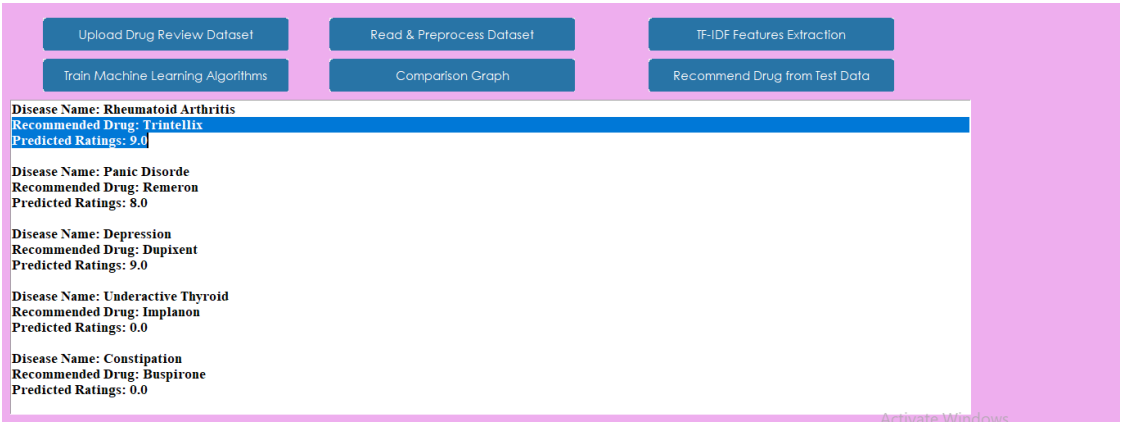


Figure 7. Drug recommendations from test data.

In above screen for each disease name application has predicted recommended drug name and ratings.