Classifying Drugs Using Machine Learning: Confidential Drugs As X,Y,Z

By

R. Sri Sushma	(20481A6042)
V. Tejaswini	(20481A6051)
P. Syam Kumar	(20481A6037)
T. Dileesh Mani	(20481A6046)

Under the guidance of

MR. K. ANIL KUMAR

Assistant Professor

ABSTRACT

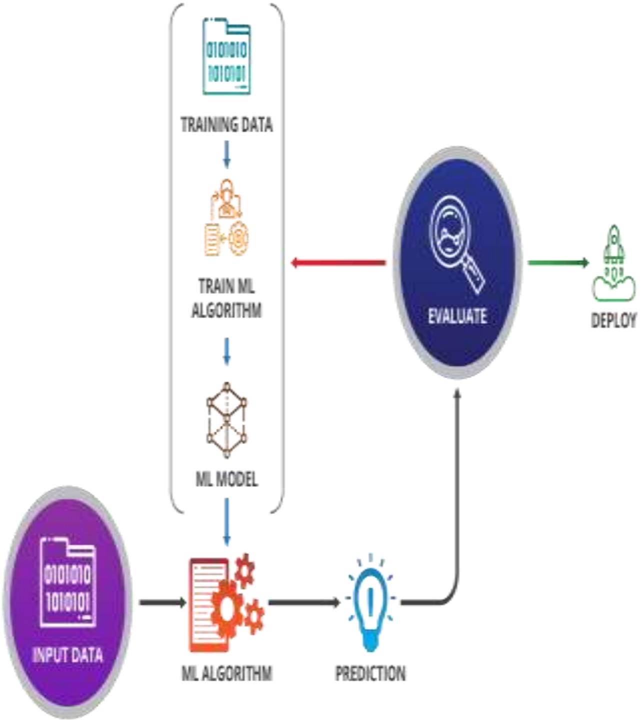
Nowadays our lifestyle has been changing. Per family, at least one person has Motorcycles or cars, etc. In the same way, we all have health issues. An earlier generation has proved “Health is Wealth”. But, for our generation, this slogan is quite challenging. We have completely moved with hybrid veggies, junk foods, etc. Due to these foods, we are not getting sufficient nutrition and suffering from health issues. To overcome this, we are consulting doctors and taking some drugs as medicines. In this project, we have some characteristics of the patients as a dataset. The target variable of this dataset is Drugs. The drug names are confidential. So, those names are replaced as Drug X, Drug Y, Drug A , Drug B, and Drug C . By consulting a doctor each time, you have to pay a doctor fee and additional charges. For saving money and time, you can use this web application to predict your drug type. The main purpose of the Drug Classification system is to predict the suitable drug type confidently for the patients based on their characteristics.We will be using classification algorithms such as Decision tree, Random forest, KNN, and xgboost. We will train and test the data with these algorithms. From this best model is selected and saved in pkl format. We will be doing flask integration.

OBJECTIVES

- The primary objective of this research is to develop an advanced and personalized drug classification model that takes into account various critical factors, including age, sex, blood pressure, cholesterol levels, and NatoK.
- To know , which drug is suitable for the patients based on their health condition.

METHODOLOGY

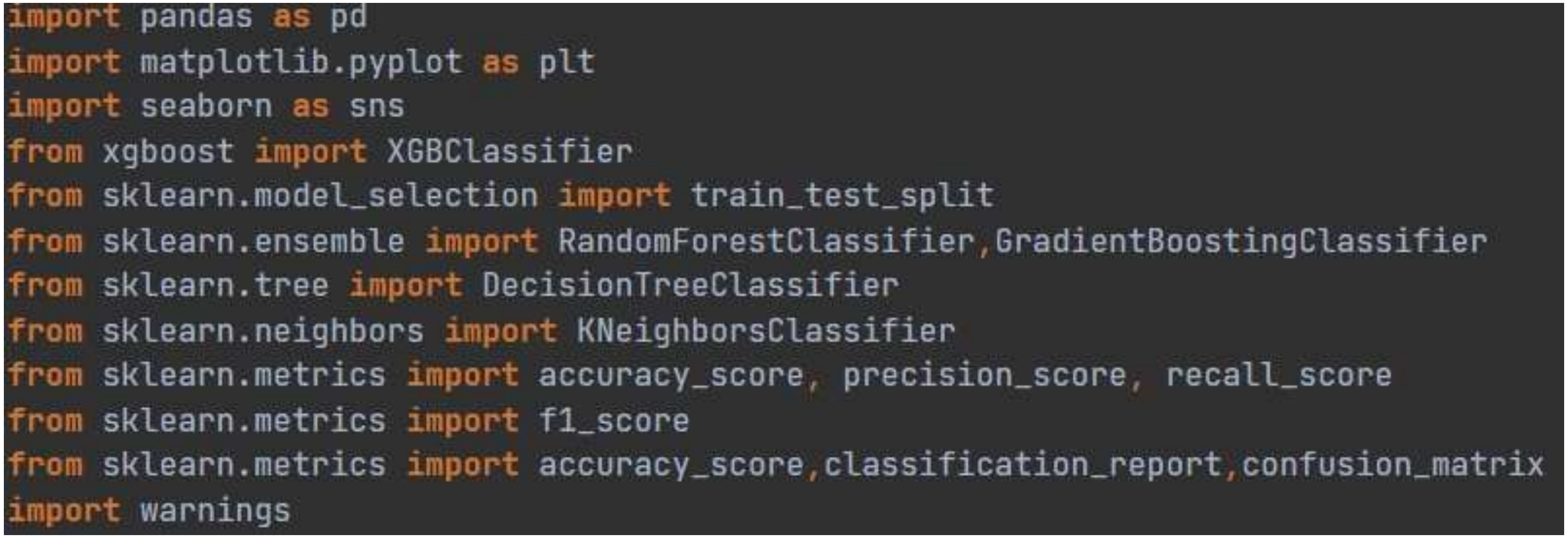
- The classification of drugs can be done by using Machine learning algorithms. And here we used RandomForest model to predict the accurate value.
- The steps to be followed to build the model are :
  a. Dataset
  b. Data preprocessing
  c. Model training
  d. Prediction



- Using the Flask UI to deploy the model and implementation of our application.

REQUIREMENTS

- Jupyter Notebook / Pycharm / VS code
- Dataset (drug200.csv)
- HTML-Hyper Text Markup Language
- CSS
- Flask

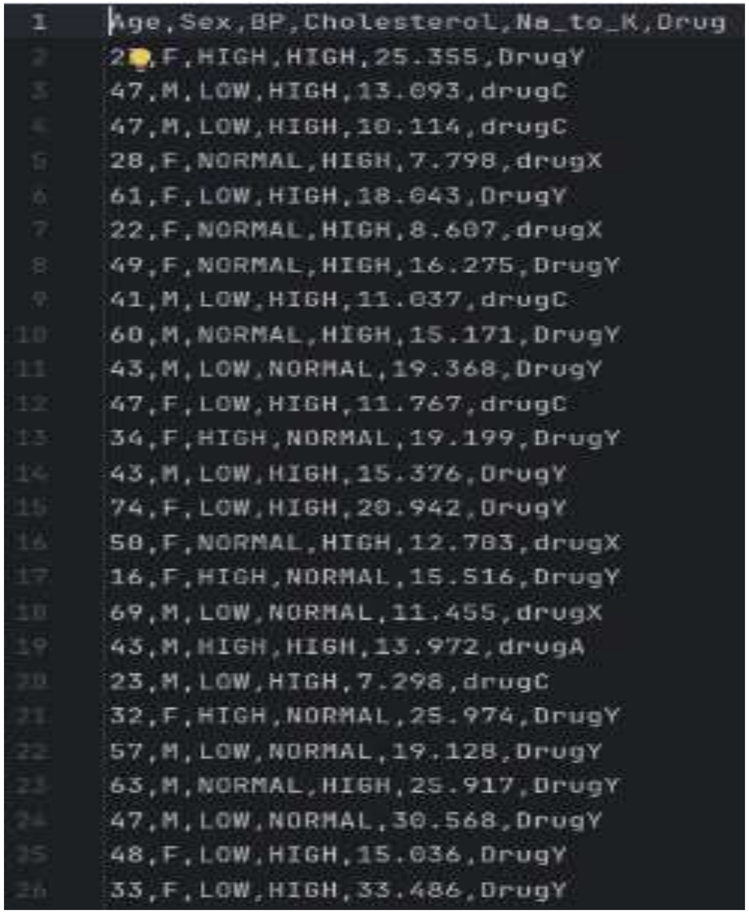


```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
import warnings
```

IMPLEMENTATION

The proposed system is implemented using Jupyter Notebook and Flask UI. Steps included in this implementation are Data gathering , preprocessing, model training and prediction.

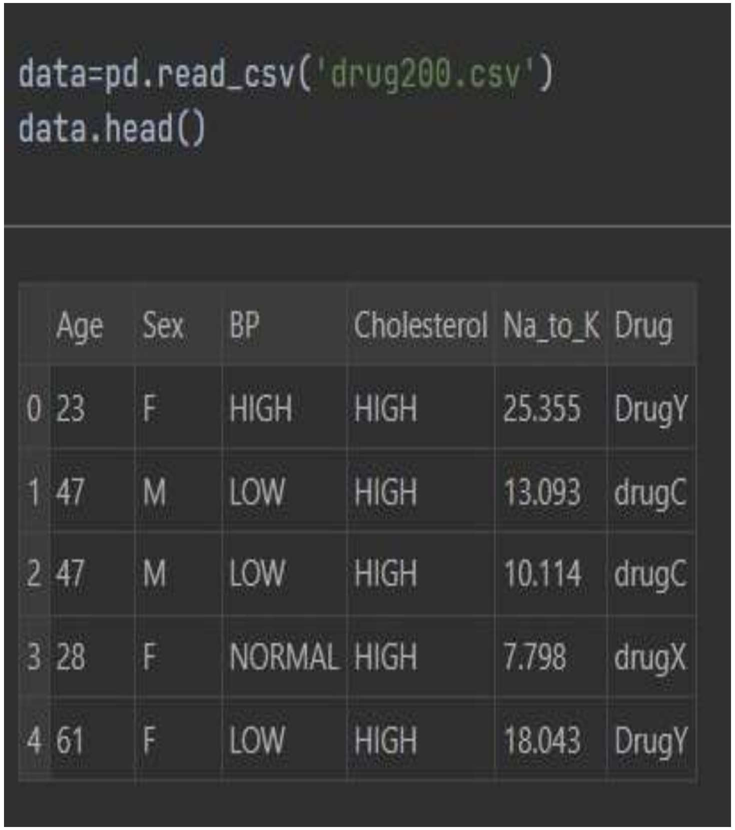
Data Gathering : The data was sourced from the kaggle which contains 200 patients data with about five attributes and one target attribute that is Drug.



```
Age,Sex,BP,Cholesterol,Na_to_K,Drug
2[illegible],F,HIGH,HIGH,25.355,DrugY
47,M,LOW,HIGH,13.093,drugC
47,M,LOW,HIGH,10.114,drugC
28,F,NORMAL,HIGH,7.798,drugX
61,F,LOW,HIGH,18.043,DrugY
22,F,NORMAL,HIGH,8.607,drugX
49,F,NORMAL,HIGH,16.275,DrugY
41,M,LOW,HIGH,11.037,drugC
60,M,NORMAL,HIGH,15.171,DrugY
43,M,LOW,NORMAL,19.368,DrugY
47,F,LOW,HIGH,11.767,drugC
34,F,HIGH,NORMAL,19.199,DrugY
43,M,LOW,HIGH,15.376,DrugY
74,F,LOW,HIGH,20.942,DrugY
50,F,NORMAL,HIGH,12.703,drugX
16,F,HIGH,NORMAL,15.516,DrugY
69,M,LOW,NORMAL,11.455,drugX
43,M,HIGH,HIGH,13.972,drugA
23,M,LOW,HIGH,7.298,drugC
32,F,HIGH,NORMAL,25.974,DrugY
57,M,LOW,NORMAL,19.128,DrugY
63,M,NORMAL,HIGH,25.917,DrugY
47,M,LOW,NORMAL,30.568,DrugY
48,F,LOW,HIGH,15.036,DrugY
33,F,LOW,HIGH,33.486,DrugY
```

Data Preprocessing: The collected data undergoes preprocessing to clean and prepare it for model training. This step involves handling missing values, normalizing features, and dealing with class imbalance, if present.

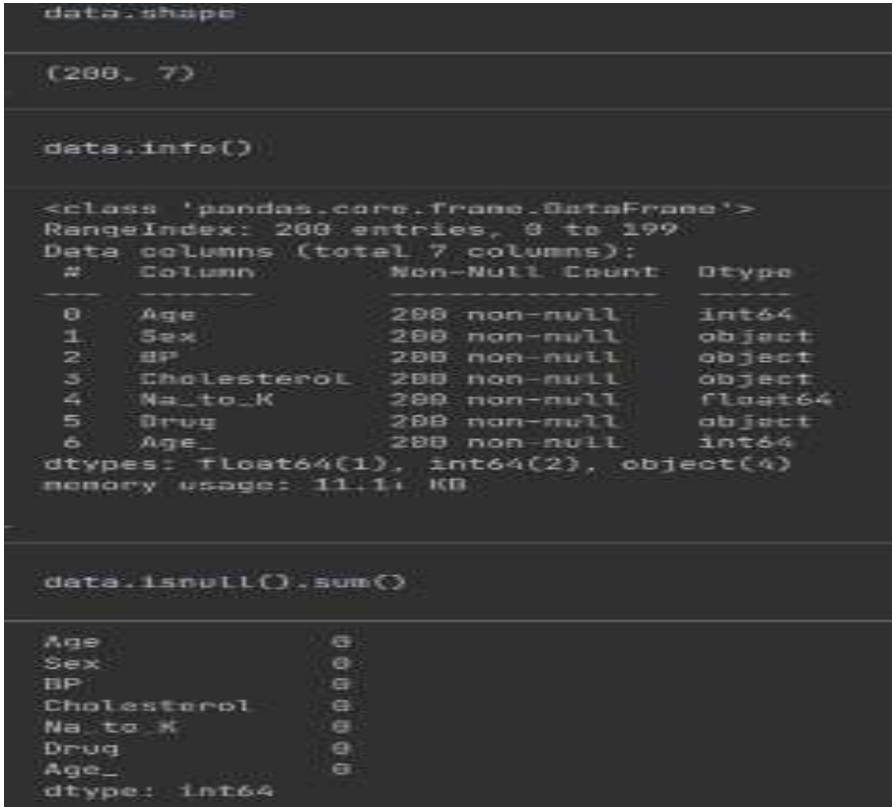
- Reading data from the dataset:



```
data=pd.read_csv('drug200.csv')
data.head()
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

- Data preprocessing:



```
data.shape
```

```
(200, 7)
```

```
data.info()
```

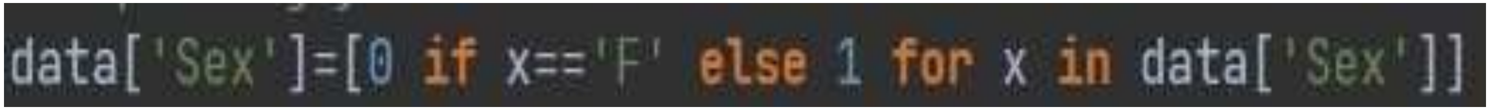
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Age          200 non-null    int64
 1   Sex          200 non-null    object
 2   BP           200 non-null    object
 3   Cholesterol  200 non-null    object
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object
 6   Age_         200 non-null    int64
dtypes: float64(1), int64(2), object(4)
memory usage: 11.1+ KB
```

```
data.isnull().sum()
```

```
Age            0
Sex            0
BP             0
Cholesterol    0
Na_to_K        0
Drug           0
Age_           0
dtype: int64
```

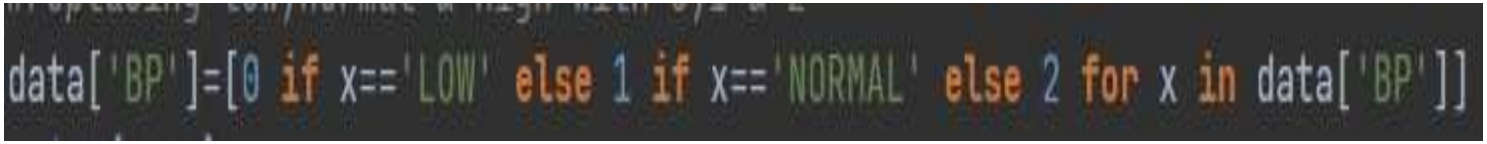
- Performing the Label Encoding on categorical variables.

-handling the sex:



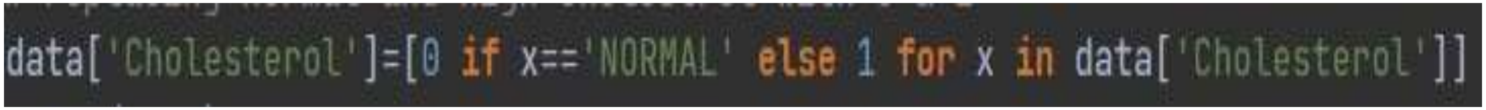
```
data['Sex']=[0 if x=='F' else 1 for x in data['Sex']]
```

-handling the BP :



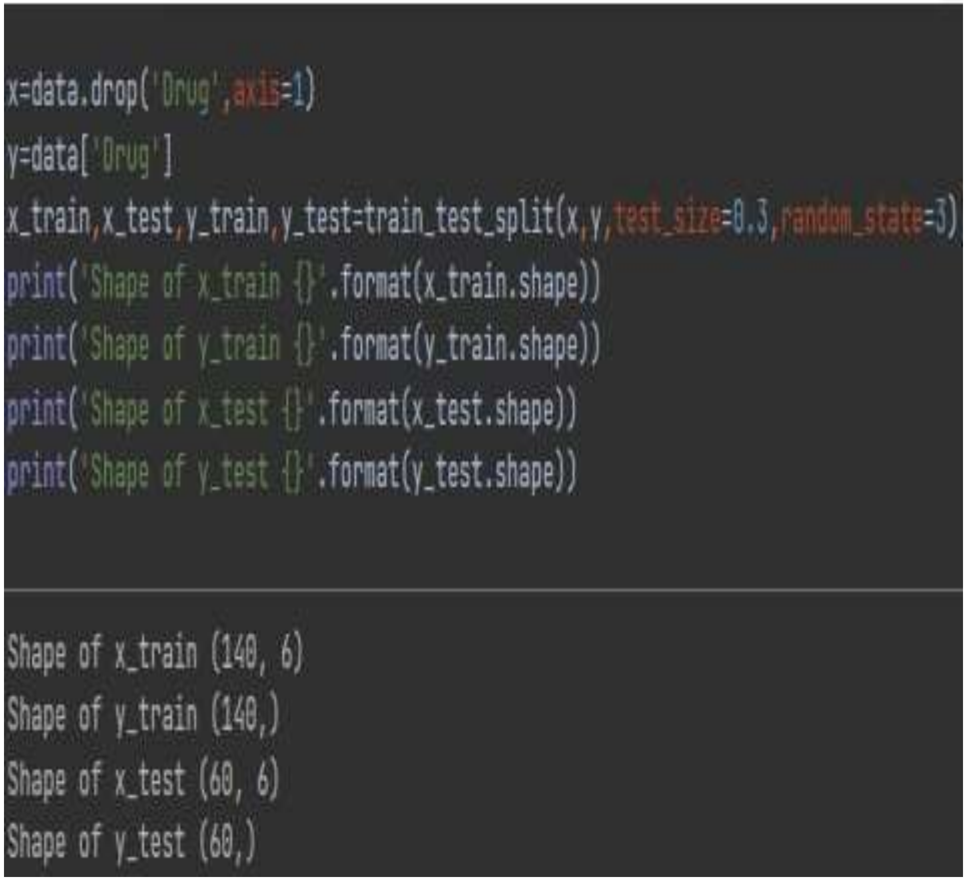
```
data['BP']=[0 if x=='LOW' else 1 if x=='NORMAL' else 2 for x in data['BP']]
```

-handling the Cholesterol:



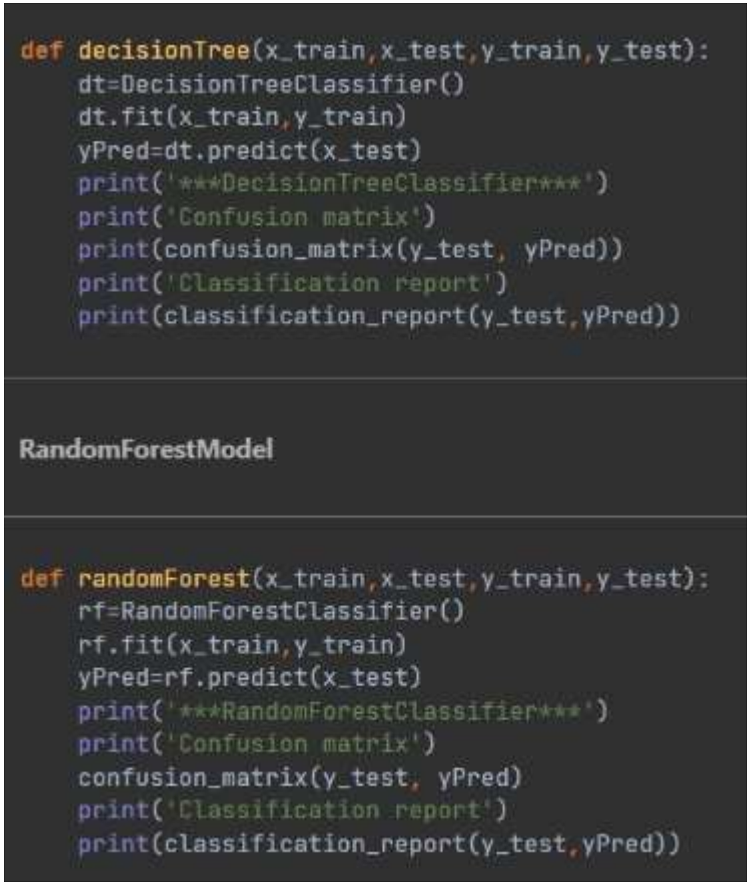
```
data['Cholesterol']=[0 if x=='NORMAL' else 1 for x in data['Cholesterol']]
```

Model Training: Various machine learning algorithms, such as SVM, Random Forest, and decision tree, are evaluated for their performance in drug classification.



```python
x=data.drop('Drug',axis=1)
y=data['Drug']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=3)
print('Shape of x_train {}'.format(x_train.shape))
print('Shape of y_train {}'.format(y_train.shape))
print('Shape of x_test {}'.format(x_test.shape))
print('Shape of y_test {}'.format(y_test.shape))
```

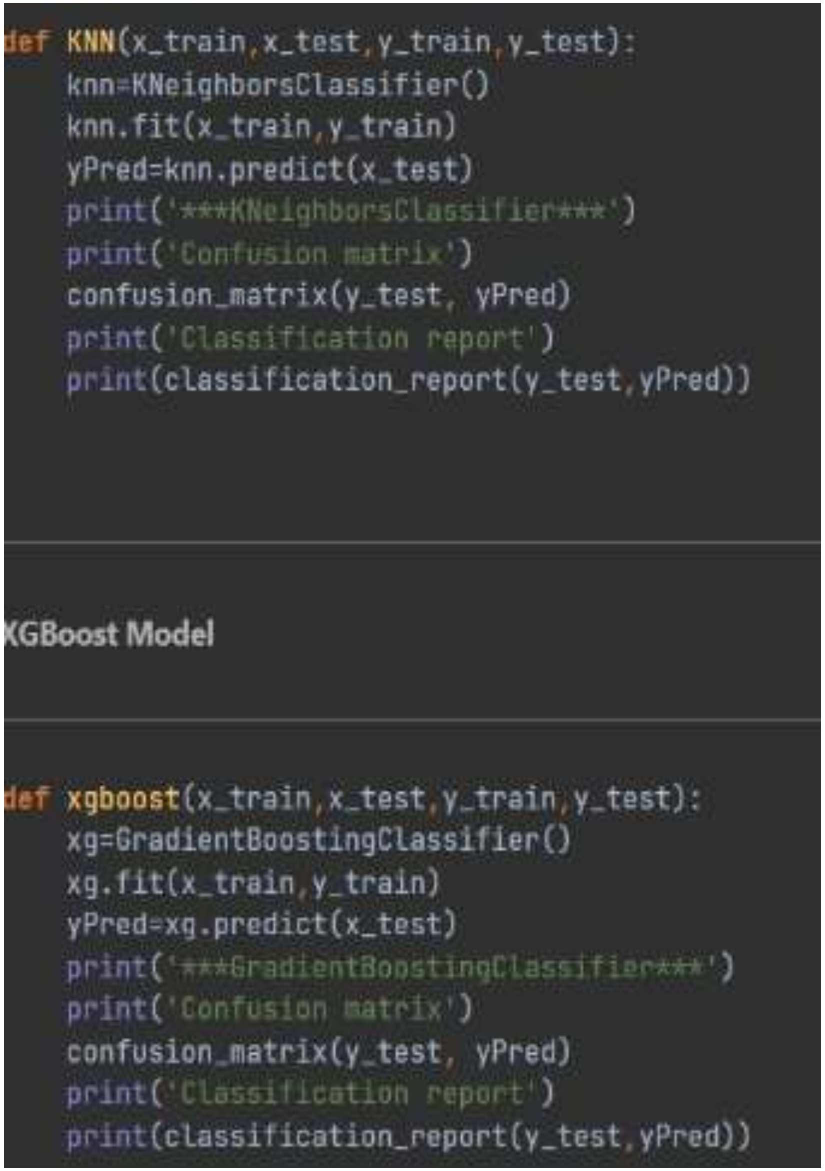
```
Shape of x_train (140, 6)
Shape of y_train (140,)
Shape of x_test (60, 6)
Shape of y_test (60,)
```



```python
def decisionTree(x_train,x_test,y_train,y_test):
    dt=DecisionTreeClassifier()
    dt.fit(x_train,y_train)
    yPred=dt.predict(x_test)
    print('***DecisionTreeClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test, yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

RandomForestModel

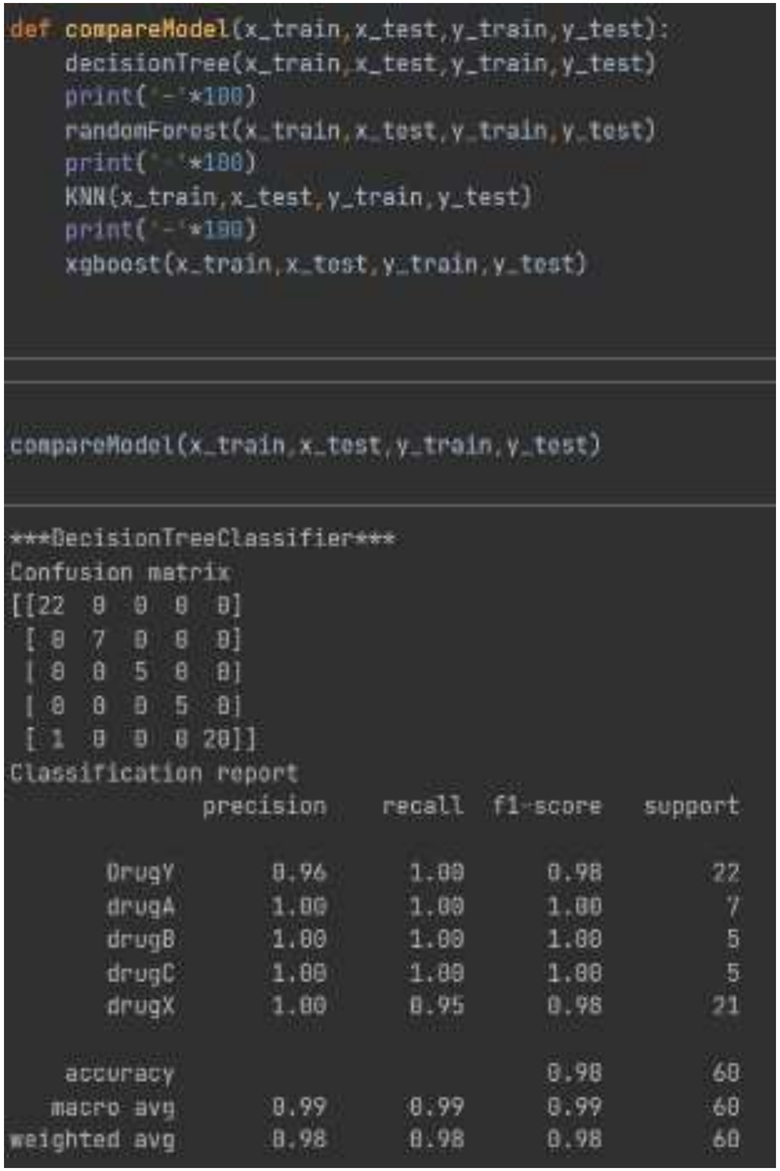
```python
def randomForest(x_train,x_test,y_train,y_test):
    rf=RandomForestClassifier()
    rf.fit(x_train,y_train)
    yPred=rf.predict(x_test)
    print('***RandomForestClassifier***')
    print('Confusion matrix')
    confusion_matrix(y_test, yPred)
    print('Classification report')
    print(classification_report(y_test,yPred))
```



```
def KNN(x_train,x_test,y_train,y_test):
    knn=KNeighborsClassifier()
    knn.fit(x_train,y_train)
    yPred=knn.predict(x_test)
    print('***KNeighborsClassifier***')
    print('Confusion matrix')
    confusion_matrix(y_test, yPred)
    print('Classification report')
    print(classification_report(y_test,yPred))
```

XGBoost Model

```
def xgboost(x_train,x_test,y_train,y_test):
    xg=GradientBoostingClassifier()
    xg.fit(x_train,y_train)
    yPred=xg.predict(x_test)
    print('***GradientBoostingClassifier***')
    print('Confusion matrix')
    confusion_matrix(y_test, yPred)
    print('Classification report')
    print(classification_report(y_test,yPred))
```



```
def compareModel(x_train,x_test,y_train,y_test):
    decisionTree(x_train,x_test,y_train,y_test)
    print('-'*100)
    randomForest(x_train,x_test,y_train,y_test)
    print('-'*100)
    KNN(x_train,x_test,y_train,y_test)
    print('-'*100)
    xgboost(x_train,x_test,y_train,y_test)
```

```
compareModel(x_train,x_test,y_train,y_test)
```

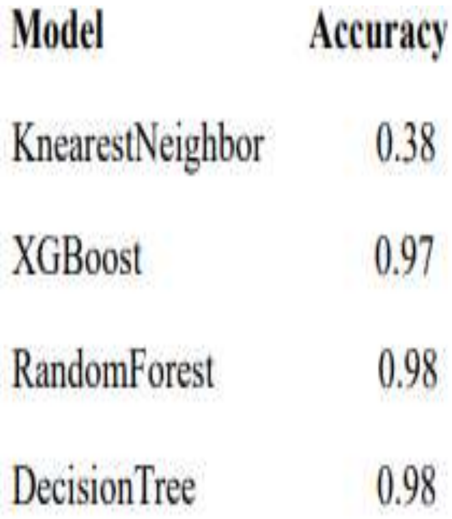
```
***DecisionTreeClassifier***
Confusion matrix
[[22  0  0  0  0]
 [ 0  7  0  0  0]
 [ 0  0  5  0  0]
 [ 0  0  0  5  0]
 [ 1  0  0  0 20]]
Classification report
              precision    recall  f1-score   support

       DrugY       0.96      1.00      0.98        22
       drugA       1.00      1.00      1.00         7
       drugB       1.00      1.00      1.00         5
       drugC       1.00      1.00      1.00         5
       drugX       1.00      0.95      0.98        21

    accuracy                           0.98        60
   macro avg       0.99      0.99      0.99        60
weighted avg       0.98      0.98      0.98        60
```

Prediction : Using the model we predict the output values for the given data.

Accuracy Score :Using the random Forest and decision tree we got the accuracy score about 98% or 0.98.

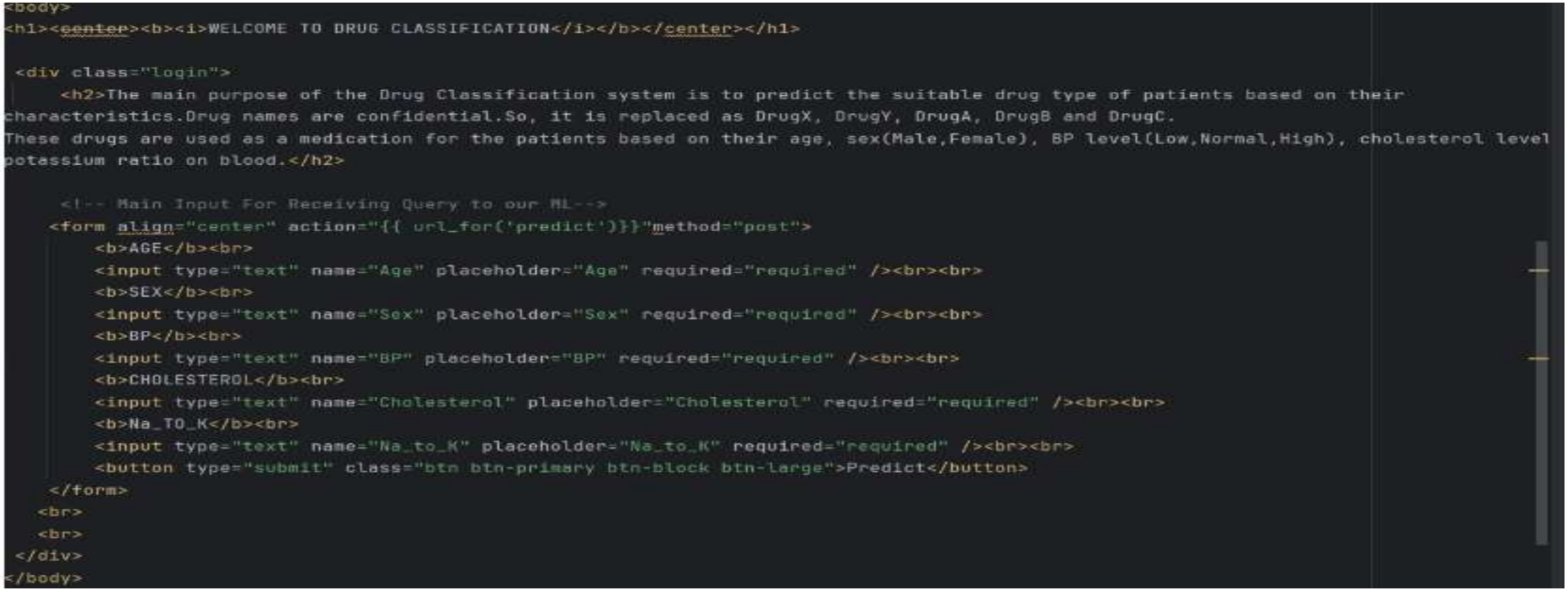


Model	Accuracy
KnearestNeighbor	0.38
XGBoost	0.97
RandomForest	0.98
DecisionTree	0.98

Flask:

- Flask is a web framework that provides libraries to build web applications in python.
- We can now install the flask by using the following command.

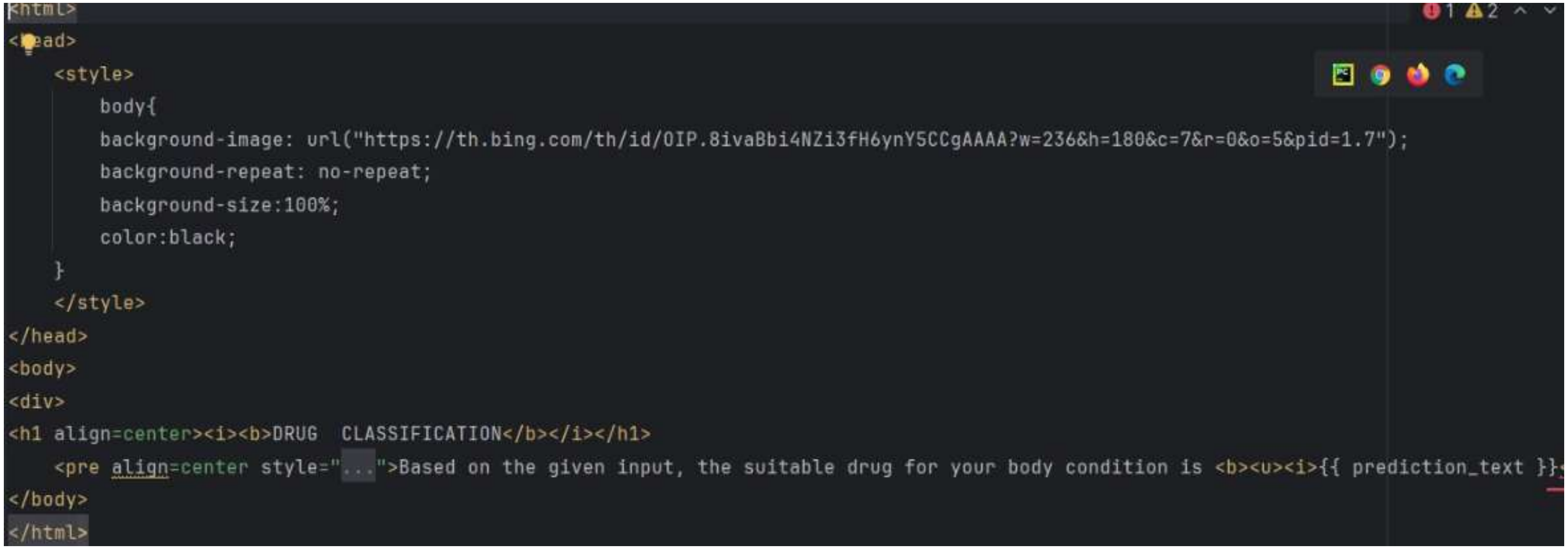
$ pip install flask



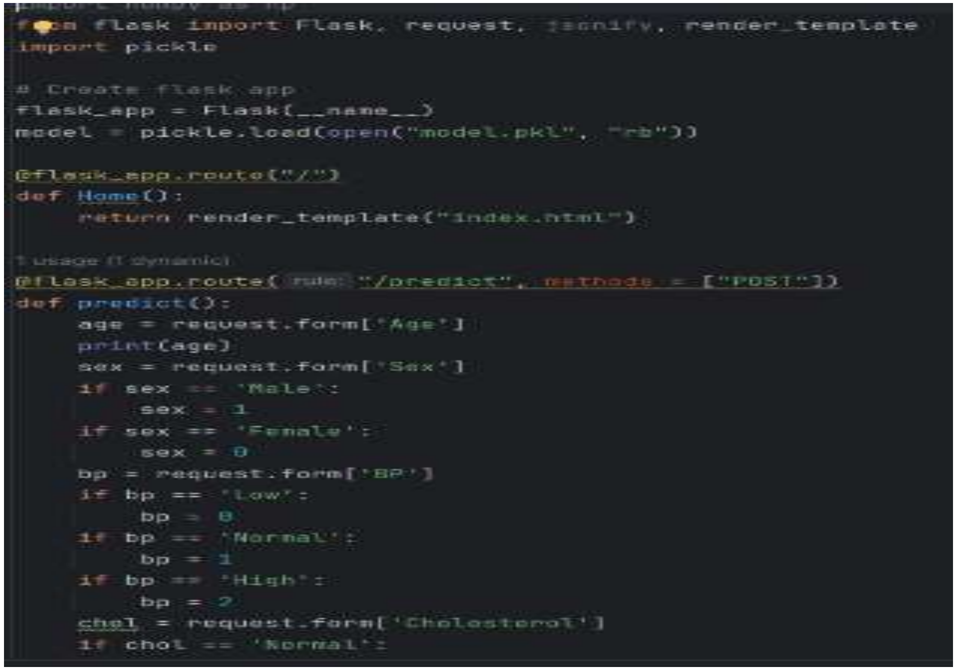
```
<body>
<h1><center><b><i>WELCOME TO DRUG CLASSIFICATION</i></b></center></h1>

 <div class="login">
     <h2>The main purpose of the Drug Classification system is to predict the suitable drug type of patients based on their
characteristics.Drug names are confidential.So, it is replaced as DrugX, DrugY, DrugA, DrugB and DrugC.
These drugs are used as a medication for the patients based on their age, sex(Male,Female), BP level(Low,Normal,High), cholesterol level
potassium ratio on blood.</h2>

     <!-- Main Input For Receiving Query to our ML-->
    <form align="center" action="{{ url_for('predict')}}"method="post">
        <b>AGE</b><br>
        <input type="text" name="Age" placeholder="Age" required="required" /><br><br>
        <b>SEX</b><br>
        <input type="text" name="Sex" placeholder="Sex" required="required" /><br><br>
        <b>BP</b><br>
        <input type="text" name="BP" placeholder="BP" required="required" /><br><br>
        <b>CHOLESTEROL</b><br>
        <input type="text" name="Cholesterol" placeholder="Cholesterol" required="required" /><br><br>
        <b>Na_TO_K</b><br>
        <input type="text" name="Na_to_K" placeholder="Na_to_K" required="required" /><br><br>
        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>
   <br>
   <br>
 </div>
</body>
```



```
<html>
<head>
    <style>
        body{
        background-image: url("https://th.bing.com/th/id/OIP.8ivaBbi4NZi3fH6ynY5CCgAAAA?w=236&h=180&c=7&r=0&o=5&pid=1.7");
        background-repeat: no-repeat;
        background-size:100%;
        color:black;
    }
    </style>
</head>
<body>
<div>
<h1 align=center><i><b>DRUG  CLASSIFICATION</b></i></h1>
    <pre align=center style="...">Based on the given input, the suitable drug for your body condition is <b><u><i>{{ prediction_text }}
</body>
</html>
```



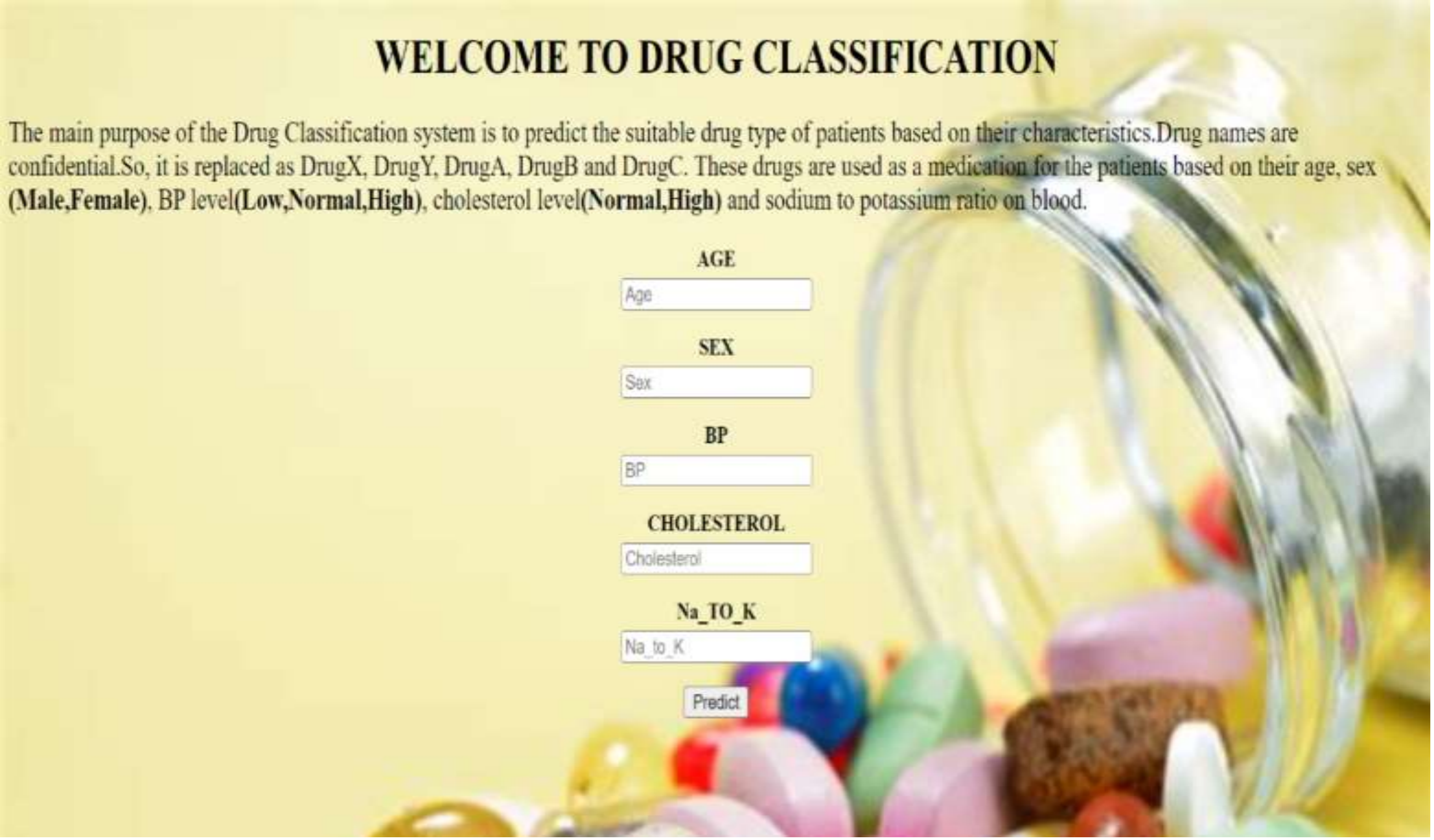
```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

# Create flask app
flask_app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))

@flask_app.route("/")
def Home():
    return render_template("index.html")

1 usage (1 dynamic)
@flask_app.route( rule: "/predict", methods = ["POST"])
def predict():
    age = request.form['Age']
    print(age)
    sex = request.form['Sex']
    if sex == 'Male':
        sex = 1
    if sex == 'Female':
        sex = 0
    bp = request.form['BP']
    if bp == 'Low':
        bp = 0
    if bp == 'Normal':
        bp = 1
    if bp == 'High':
        bp = 2
    chol = request.form['Cholesterol']
    if chol == 'Normal':
```

Output Screen Shots



WELCOME TO DRUG CLASSIFICATION

The main purpose of the Drug Classification system is to predict the suitable drug type of patients based on their characteristics.Drug names are confidential.So, it is replaced as DrugX, DrugY, DrugA, DrugB and DrugC. These drugs are used as a medication for the patients based on their age, sex **(Male,Female)**, BP level**(Low,Normal,High)**, cholesterol level**(Normal,High)** and sodium to potassium ratio on blood.

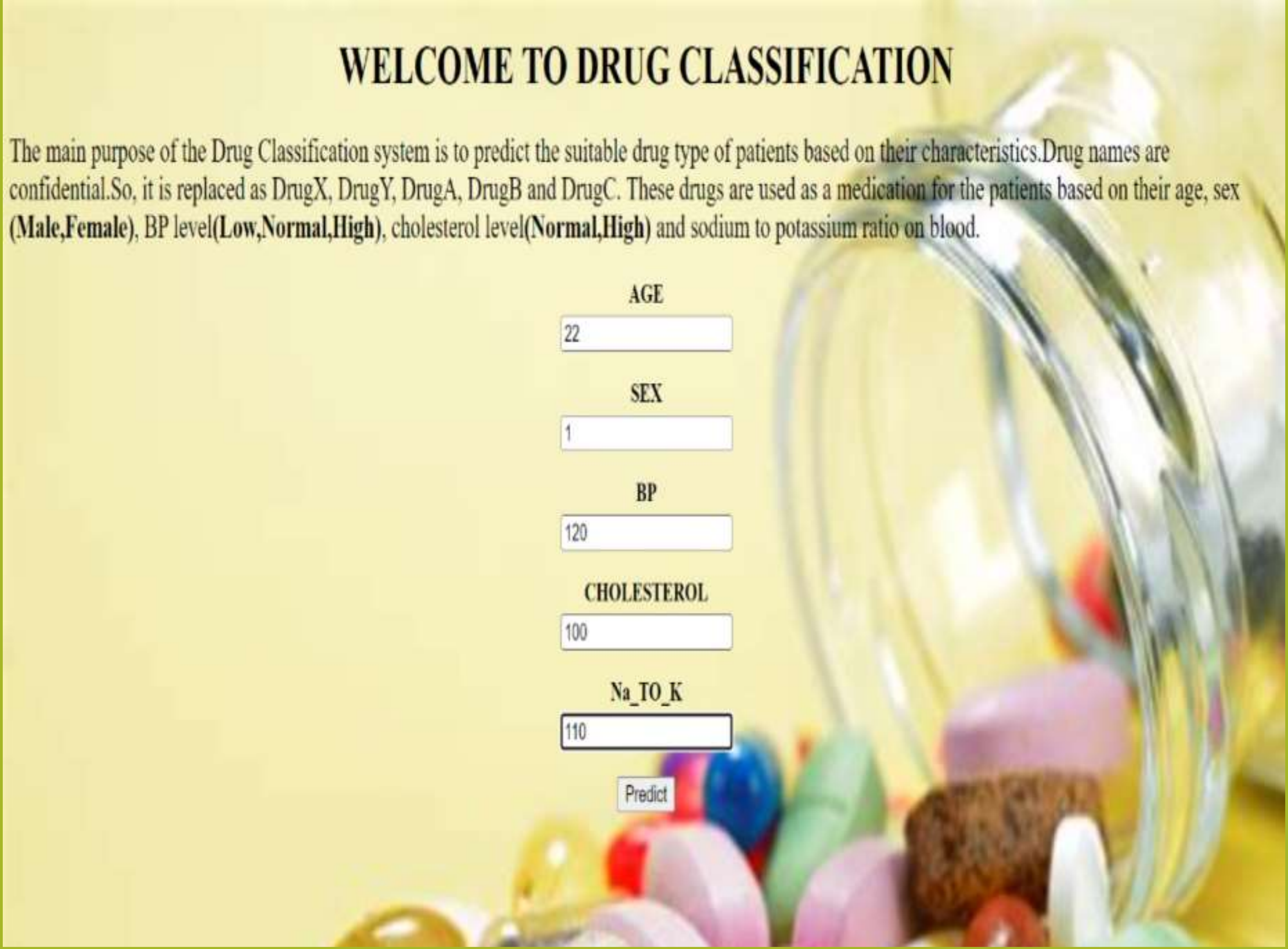
AGE: Age

SEX: Sex

BP: BP

CHOLESTEROL: Cholesterol

Na_TO_K: Na_to_K



WELCOME TO DRUG CLASSIFICATION

The main purpose of the Drug Classification system is to predict the suitable drug type of patients based on their characteristics.Drug names are confidential.So, it is replaced as DrugX, DrugY, DrugA, DrugB and DrugC. These drugs are used as a medication for the patients based on their age, sex **(Male,Female)**, BP level**(Low,Normal,High)**, cholesterol level**(Normal,High)** and sodium to potassium ratio on blood.

AGE

22

SEX

1

BP

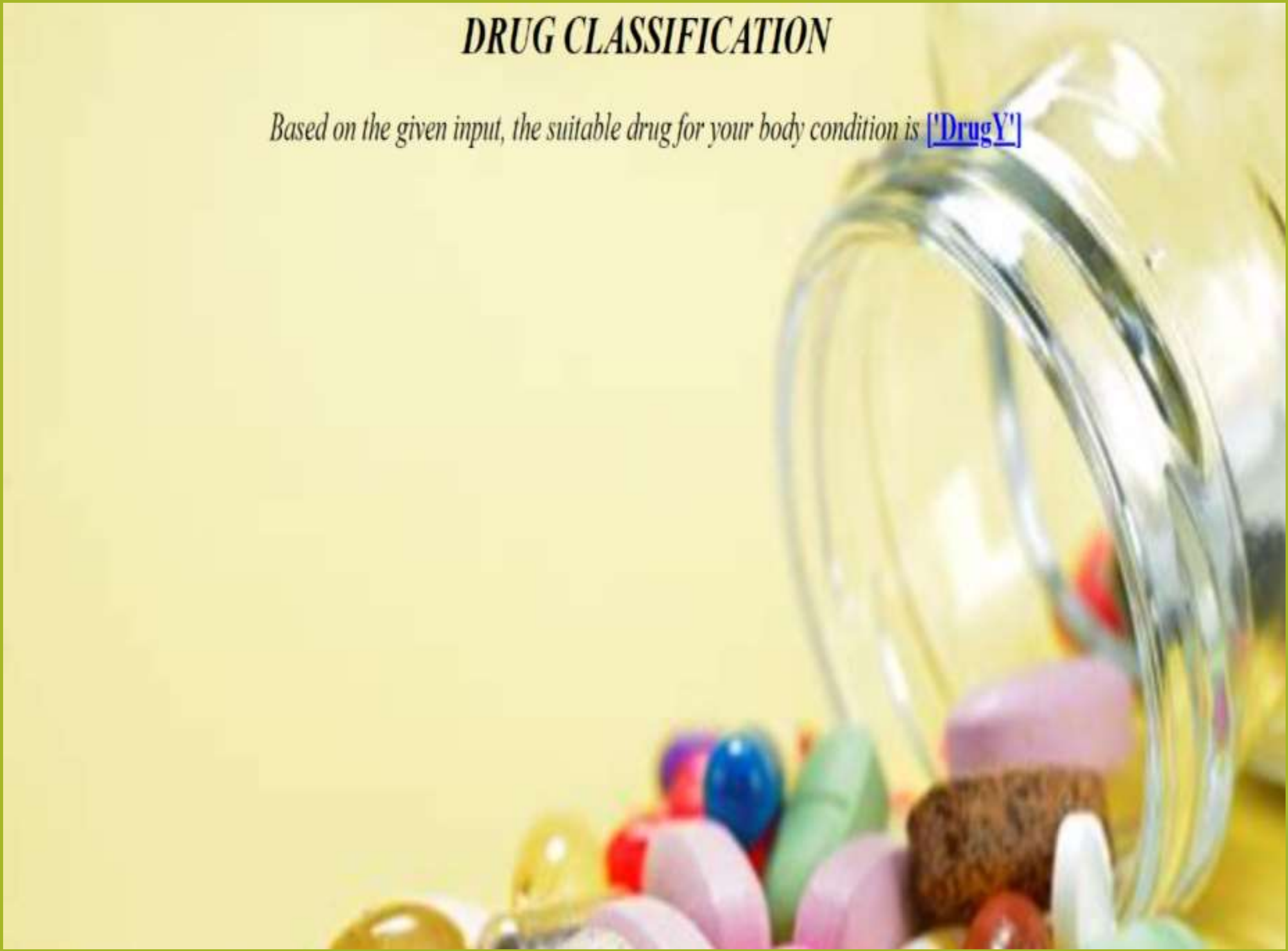
120

CHOLESTEROL

100

Na_TO_K

110



DRUG CLASSIFICATION

Based on the given input, the suitable drug for your body condition is ['DrugY']

Thank you