Drug Classification Using Machine Learning

1. INTRODUCTION

Overview

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 DrugX, DrugY, DrugA , DrugB, and DrugC . 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. The main problem here is not just the feature sets and target sets but also the approach that is taken in solving these types of problems.

Proposed System

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 and IBM deployment.

2. Read The Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of csv file.

2.1 Univariate Analysis

In simple words, univariate analysis is understanding the data with single feature. Here we have displayed two different graphs such as distplot and countplot.

- Seaborn package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot.

- In our dataset we have some categorical features. With the countplot function, we are going to count the unique category in those features. We have created a dummy data frame with categorical features. With for loop and subplot we have plotted this below graph.

- From the plot we came to know, Most of the patients are using drugY and drugX. And most of the patients have high BP and high Cholesterol.

2.2 Bivariate Analysis

To find the relation between two features we use bivariate analysis. Here we are visualizing the relationship between drug & BP, drug & sex and drug & cholesterol.

- Countplot is used here. As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.
- From the below plot you can understand that drugA and drugB is not preferred to low and normal BP patients. DrugC is preferred only to low BP patients.
- By third graph we can understand, drugC is not preferred to normal cholesterol patients.

With the help of age feature we are creating an age interval and finding the relation between drug

feature and age interval feature. Function crosstab is used to find the relationship. From the below image we get a clear understanding, DrugB is preferred only for patients above age 50 years. And drugA is not preferred for patients above age 50 years.

2.3 Multivariate Analysis

In simple words, multivariate analysis is to find the relation between multiple features.

Here we have used swarmplot from seaborn package.

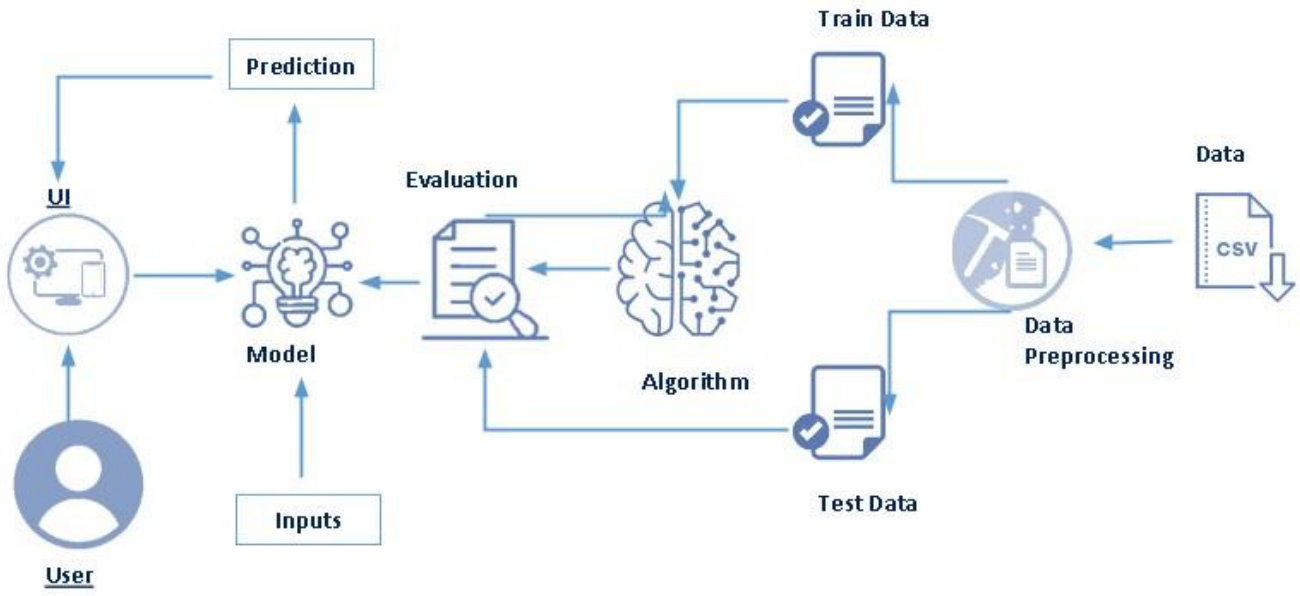
- From the below image, we came to a conclusion that DrugY is used by most of patients who has different BP levels. But It is preferred only for patients having Na_to_K > 15 (Na_to_K – Sodium to potassium ratio on blood).

2.4 Descriptive Analysis

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

3. THE ORETICAL ANALYSIS

3.1 Block Diagram



3.2 Pre-Requisites

Anaconda Navigator

Python packages

Flask - Web framework used for building Web applications.

- Type **"pip install numpy "** and click enter
- Type **"pip install pandas "** and click enter
- Type **"pip install scikit-learn"** and click enter.
- Type **"pip install matplotlib"** and click enter.
- Type **"pip install scipy"** and click enter.
- Type **"pip install pickle-mixin"** and click enter.
- Type **"pip install seaborn"** and click enter.

- Type **“pip install Flask ”** and click enter.

4 Prior Knowledge

One should have knowledge of the following Concepts

Please refer to the videos below to gain sufficient required knowledge to complete the project.

- **Supervised and unsupervised learning**
- **Regression Classification and Clustering**
- **Random Forest Classifier**
- **Ensemble Technique**
- **Decision Tree Classifier**
- **NN**: Refer the link
- **Xgboost**: Refer the link
- **Evaluation metrics:** Refer the link
- **Flask**

5. Project Flow

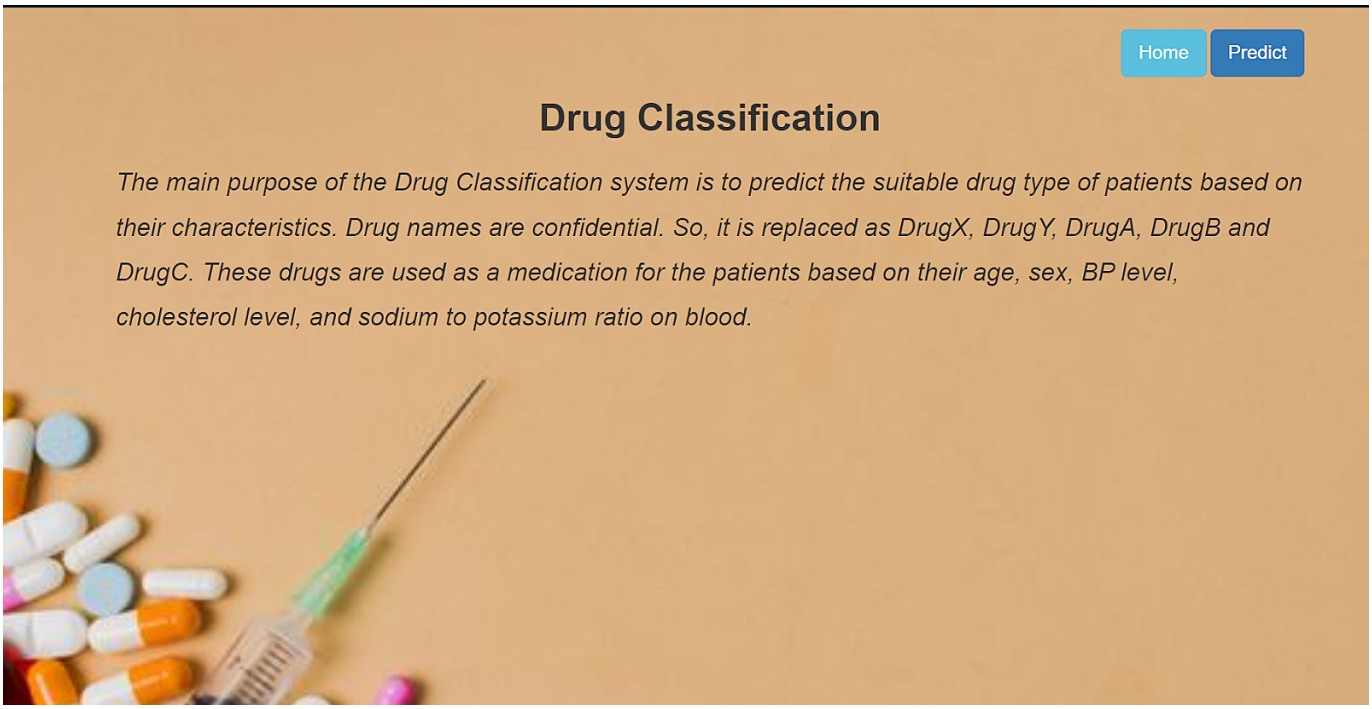
Project Flow:

- The user interacts with the UI to enter the input.
- Entered input is analyzed by the model which is integrated.
- Once the model analyses the input the prediction is showcased on the UI

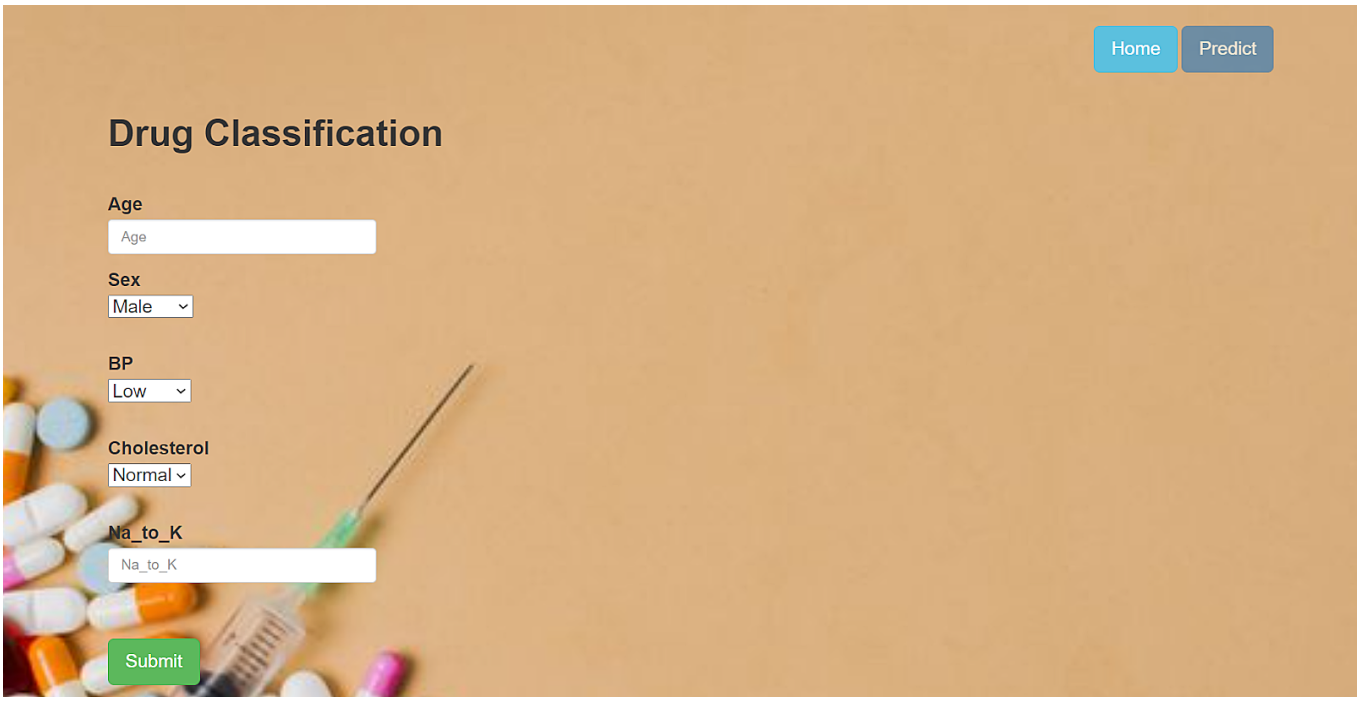
To accomplish this, we have to complete all the activities listed below,

- Data collection
    - Collect the dataset or create the dataset
- Visualizing and analyzing data
    - Univariate analysis
    - Bivariate analysis
    - Multivariate analysis
    - Descriptive analysis
- Data pre-processing
    - Checking for null values
    - Handling outlier
    - Handling categorical data
    - Splitting data into train and test
- Model building
    - Import the model building libraries
    - Initializing the model
    - Training and testing the model
    - Evaluating the performance of the model
    - Save the model
- Application Building
    - Create an HTML file
    - Build python code

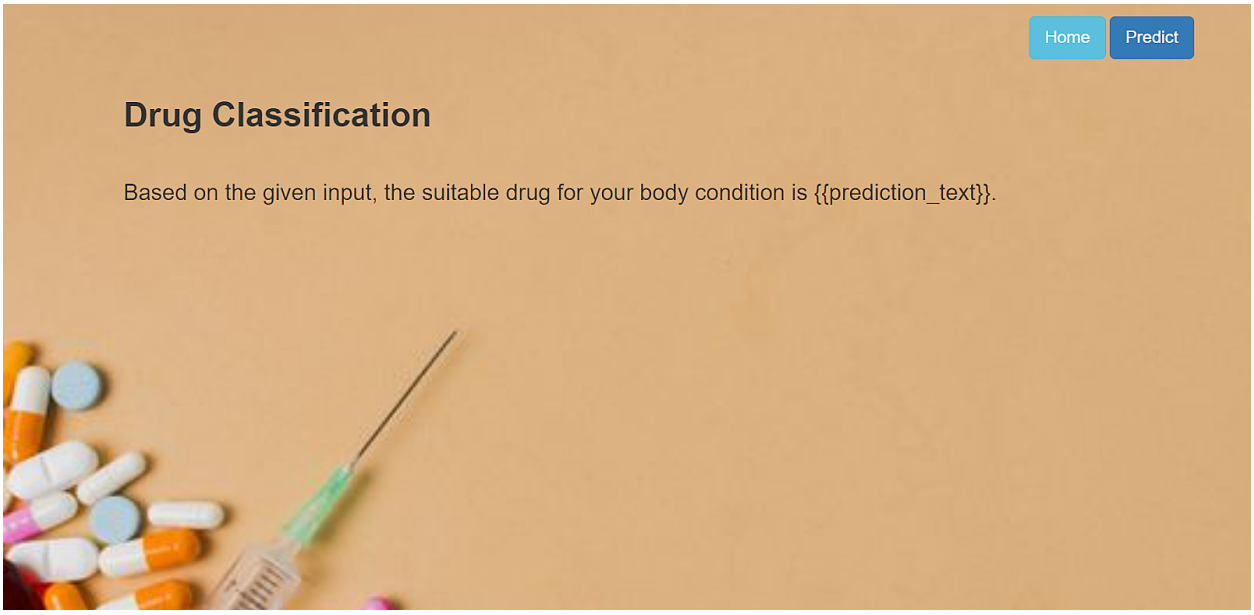
6. Result



Now when you click on predict button from top right corner you will get redirected to predict.html



Now when you click on submit button from left bottom corner you will get redirected to submit.html



7.Project Structure

- We are building a flask application that needs HTML pages stored in the templates folder and a python script app.py for scripting.
- Drug Classification.ipynb is the python file where the ML algorithm is applied to the dataset for testing and training. Finally, the model is saved for future use.
- Model.pkl is our saved model. Further, we will use this model for flask integration.
- The data folder contains the CSV file dataset for training our model.
- The training folder contains model training files and the training_ibm folder contains IBM deployment files.

8.Data Collection

ML depends heavily on data, It is most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.

9 Model Building

Now our data is cleaned and it's time to build the model. We can train our data on different algorithms. For this project we are applying four classification algorithms. The best model is saved based on its performance.

9.1Decision Tree Model

A function named decisionTree is created and train and test data are passed as the parameters. Inside the function, DecisionTreeClassifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

9.2 Random Forest Model

A function named randomForest is created and train and test data are passed as the

parameters. Inside the function, RandomForestClassifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

9.3 NN Model

A function named KNN is created and train and test data are passed as the parameters. Inside the function, KNeighborsClassifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

9.4 Xgboost Model

A function named xgboost is created and train and test data are passed as the parameters. Inside the function, GradientBoostingClassifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

10 Run The Application

Run the application

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type “python app.py” command
- Navigate to the localhost where you can view your web page.
- Click on the predict button from the top right corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

11 Project Objectives

By the end of this project:

- You’ll be able to understand the problem to classify if it is a regression or a classification kind of problem.

- You will be able to know how to pre-process/clean the data using different data preprocessing techniques.
- You will be able to analyze or get insights into data through visualization.
- Applying different algorithms according to the dataset and based on visualization.
- You will be able to know how to build a web application using the Flask framework

12. APPENDIX

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jun  8 10:12:54 2022

@author: user
"""
from flask import Flask, render_template, request
import numpy as np
import pickle
import pandas as pd

model = pickle.load(open(r"C:/Users/DELL/Desktop/Drug Pro/Flask/model.pkl",'rb'))
```

```
app = Flask(__name__)



@app.route("/")
def about():
    return render_template('home.html')




@app.route("/predict")
def home1():
    return render_template('predict.html')



@app.route("/submit")
def home2():
    return render_template('submit.html')



if __name__ == "__main__":
 app.run(debug=False)
```