Project Report

On

Drug Classification Model

Submitted in partial fulfilment of the requirements for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

(Artificial Intelligence & Machine Learning)

by

Ms. G TEJASWINI (22WH1A6608)

Ms. G ANUSHA (22WH1A6641)

Ms. G VAISHNAVI (22WH1A6658)

Ms. B ANITHA (22WH1A6659)

Under the esteemed guidance of
Ms. A Naga Kalyani
Assistant Professor, CSE(AI&ML)

Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with A Grade

Bachupally, Hyderabad – 500090

2024-25

Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with A Grade

Bachupally, Hyderabad – 500090

2024-25

CERTIFICATE

This is to certify that the major project entitled **"Drug Classification Model"** is a bonafide work carried out by **Ms. G. Tejaswini (22WH1A6608), Ms. G. Anusha (22WH1A6641)**, **Ms. G. Vaishnavi (22WH1A6658), Ms. B. Anitha (22WH1A6659**) in partial fulfilment for the award of B. Tech degree in **Computer Science & Engineering (AI&ML)**, **BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision. The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Supervisor

Ms. A Naga Kalyani
Assistant Professor
Dept of CSE(AI&ML)

Head of the Department

Dr. B. Lakshmi Praveena
HOD & Professor
Dept of CSE(AI&ML)

External Examiner

DECLARATION

We hereby declare that the work presented in this project entitled **"Drug Classification Model"** submitted towards completion of Project work in III Year of B.Tech of CSE(AI&ML) at **BVRIT HYDERABAD College of Engineering for Women,** Hyderabad is an authentic record of our original work carried out under the guidance of **Ms. A Naga Kalyani, Assistant Professor, Department of CSE(AI&ML).**

Sign with Date:
G. Tejaswini
(22WH1A6608)

Sign with Date:
G. Anusha
(22WH1A6641)

Sign with Date:
G. Vaishnavi
(22WH1A6658)

Sign with Date:
B. Anitha
(22WH1A6659)

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. K. V. N. Sunitha, Principal, BVRIT HYDERABAD College of Engineering for Women**, for her support by providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. B. Lakshmi Praveena, Head of the Department, Department of CSE(AI&ML), BVRIT HYDERABAD College of Engineering for Women,** for all timely support and valuable suggestions during the period of our project.

We are extremely thankful to our Internal Guide, **Ms. A Naga Kalyani, Assistant Professor, CSE(AI&ML), BVRIT HYDERABAD College of Engineering for Women,** for her constant guidance and encouragement throughout the project.

Finally, we would like to thank our Major Project Coordinator, all Faculty and Staff of CSE(AI&ML) department who helped us directly or indirectly. Last but not least, we wish to acknowledge our **Parents** and **Friends** for giving moral strength and constant encouragement.

G. Tejaswini (22WH1A6608)

G. Anusha (22WH1A6641)

G. Vaishnavi (22WH1A6658)

B. Anitha (22WH1A6659)

ABSTRACT

This project focuses on developing a drug classification model that uses machine learning techniques to predict the appropriate drug for a patient based on their health parameters. The dataset used includes features such as age, sex, blood pressure, cholesterol levels, and the sodium-to-potassium ratio, with the target variable being the prescribed drug. The model employs Decision Tree and Random Forest algorithms to classify the drugs into different categories. The process involves data preprocessing, such as encoding categorical variables, followed by model training and evaluation using metrics like accuracy, precision, recall, and F1-score. The model's performance is validated using a test dataset, achieving high accuracy in drug classification. This system can assist healthcare professionals in selecting suitable medications, promoting more accurate and timely drug prescriptions while reducing human errors in decision-making.

PROBLEM STATEMENT

The goal of this project is to develop a drug classification model that predicts the therapeutic category of a drug based on patient-specific data, including age, sex, blood pressure, cholesterol levels, and sodium-to-potassium ratio. Accurate drug classification is crucial in medical practice, as it helps healthcare professionals select the most appropriate medication for patients. To achieve this, the project utilizes machine learning algorithms, specifically Decision Trees and Random Forest Classifiers, to train a model on a dataset of patient information and their corresponding drug prescriptions. The model can then predict the drug category for new patients, providing automated and reliable assistance in drug prescription.

DATASET

1	Age	Sex	BP	Cholesterol	Na_to_K	Drug
2	23	F	HIGH	HIGH	25.355	drugY
3	47	M	LOW	HIGH	13.093	drugC
4	47	M	LOW	HIGH	10.114	drugC
5	28	F	NORMAL	HIGH	7.798	drugX
6	61	F	LOW	HIGH	18.043	drugY
7	22	F	NORMAL	HIGH	8.607	drugX
8	49	F	NORMAL	HIGH	16.275	drugY
9	41	M	LOW	HIGH	11.037	drugC
10	60	M	NORMAL	HIGH	15.171	drugY
11	43	M	LOW	NORMAL	19.368	drugY
12	47	F	LOW	HIGH	11.767	drugC
13	34	F	HIGH	NORMAL	19.199	drugY
14	43	M	LOW	HIGH	15.376	drugY
15	74	F	LOW	HIGH	20.942	drugY
16	50	F	NORMAL	HIGH	12.703	drugX
17	16	F	HIGH	NORMAL	15.516	drugY
18	69	M	LOW	NORMAL	11.455	drugX
19	43	M	HIGH	HIGH	13.972	drugA
20	23	M	LOW	HIGH	7.298	drugC

DataSet Link: DataSet

Code

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn import metrics
```

Code

```
def calculate_additional_metrics(model_name, y_true, y_pred):

    precision = metrics.precision_score(y_true,
y_pred,average='weighted')

    recall = metrics.recall_score(y_true, y_pred,average='weighted')

    f1_score = metrics.f1_score(y_true, y_pred,average='weighted')


    print(f"{model_name} - Additional Metrics:")

    print(f"Precision: {precision}")

    print(f"Recall: {recall}")

    print(f"F1-Score: {f1_score}\n")
```

Code

```
ds=pd.read_csv("/content/drug200.csv")

ds.head()
```

Output

	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

Code

```
print("Description:\n",ds.describe())
```

Output

```
Description:
                Age      Na_to_K
count  200.000000  200.000000
mean    44.315000   16.084485
std     16.544315    7.223956
min     15.000000    6.269000
25%     31.000000   10.445500
50%     45.000000   13.936500
75%     58.000000   19.380000
max     74.000000   38.247000
```

Code

```
print("null values:\n",ds.isnull().sum())
```

Output

```
null values:
 Age            0
Sex             0
BP              0
Cholesterol     0
Na_to_K         0
Drug            0
dtype: int64
```

Code

```
print("data:\n",ds)
```

Output

```
data:
      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
..    ...  ..     ...         ...      ...    ...
195    56   F     LOW        HIGH   11.567  drugC
196    16   M     LOW        HIGH   12.006  drugC
197    52   M  NORMAL        HIGH    9.894  drugX
198    23   M  NORMAL      NORMAL   14.020  drugX
199    40   F     LOW      NORMAL   11.349  drugX

[200 rows x 6 columns]
```

Code

```
from sklearn import preprocessing

sex_encoding=preprocessing.LabelEncoder()

ds['Sex'] = sex_encoding.fit_transform(ds['Sex'])

BP_encoding=preprocessing.LabelEncoder()

ds['BP']=BP_encoding.fit_transform(ds['BP'])

Cholesterol_encoding=preprocessing.LabelEncoder()

ds['Cholesterol']=Cholesterol_encoding.fit_transform(ds['Cholesterol'])

#display the preprocessed dataset

print(ds)
```

Output

```
      Age  Sex  BP  Cholesterol  Na_to_K   Drug
0      23    0   0            0   25.355  drugY
1      47    1   1            0   13.093  drugC
2      47    1   1            0   10.114  drugC
3      28    0   2            0    7.798  drugX
4      61    0   1            0   18.043  drugY
..    ...  ...  ..          ...      ...    ...
195    56    0   1            0   11.567  drugC
196    16    1   1            0   12.006  drugC
197    52    1   2            0    9.894  drugX
198    23    1   2            1   14.020  drugX
199    40    0   1            1   11.349  drugX

[200 rows x 6 columns]
```

Code

```
xcols=[col for col in ds.columns if col not in ['Drug']]

x=ds[xcols]

y=ds['Drug']
```

Code

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=3)
```

Code

```
from sklearn.tree import DecisionTreeClassifier

dectree=DecisionTreeClassifier()
```

```
dectree.fit(x_train,y_train)

dr_prediction=dectree.predict(x_test)

print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_test,
dr_prediction))

calculate_additional_metrics('Decision Tree', y_test, dr_prediction)
```

Output

```
DecisionTrees's Accuracy:  0.9833333333333333
Decision Tree - Additional Metrics:
Precision: 0.9840579710144927
Recall: 0.9833333333333333
F1-Score: 0.9833152664859981
```

Code

```
from sklearn.ensemble import RandomForestClassifier

ranforest = RandomForestClassifier()

ranforest.fit(x_train, y_train)

rf_prediction= ranforest.predict(x_test)

print("Random Forest's Accuracy:", metrics.accuracy_score(y_test,
rf_prediction))

calculate_additional_metrics('Random Forest', y_test, rf_prediction)
```

Output

```
Random Forest's Accuracy: 0.9833333333333333
Random Forest - Additional Metrics:
Precision: 0.9840579710144927
Recall: 0.9833333333333333
F1-Score: 0.9833152664859981
```

Code

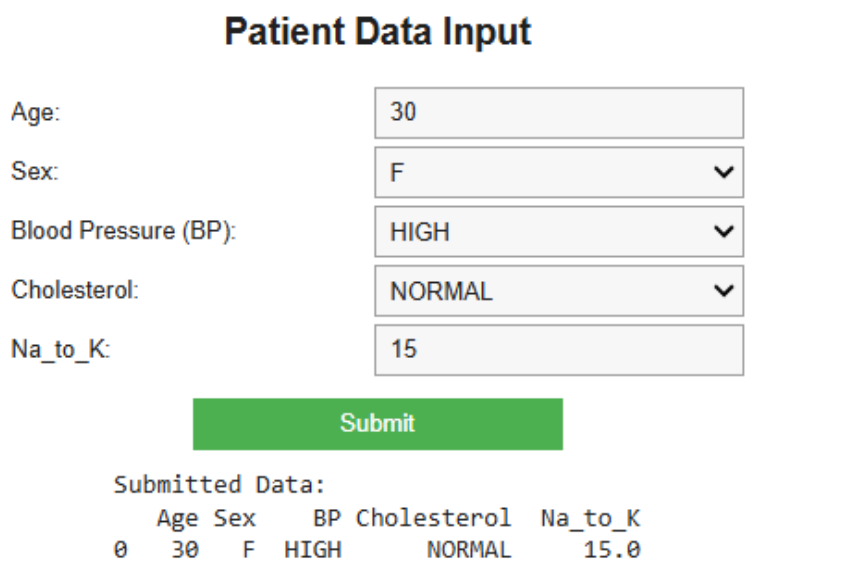
```
import pandas as pd

import ipywidgets as widgets

from IPython.display import display

# Create widgets with styled labels
```

```
age_label = widgets.Label(value="Age:", style={'font_size': '16px'})

age_widget = widgets.IntText(value=0,
layout=widgets.Layout(width='200px'))


sex_label = widgets.Label(value="Sex:", style={'font_size': '16px'})

sex_widget = widgets.Dropdown(options=["M", "F"],
layout=widgets.Layout(width='200px'))

bp_label = widgets.Label(value="Blood Pressure (BP):",
style={'font_size': '16px'})

bp_widget = widgets.Dropdown(options=["LOW", "NORMAL", "HIGH"],
layout=widgets.Layout(width='200px'))

cholesterol_label = widgets.Label(value="Cholesterol:",
style={'font_size': '16px'})

cholesterol_widget = widgets.Dropdown(options=["NORMAL", "HIGH"],
layout=widgets.Layout(width='200px'))

na_to_k_label = widgets.Label(value="Na_to_K:", style={'font_size':
'16px'})

na_to_k_widget = widgets.FloatText(value=0.0,
layout=widgets.Layout(width='200px'))

submit_button = widgets.Button(description="Submit",

                               button_style='success',  # 'success',
'info', 'warning', 'danger', or ''

                               layout=widgets.Layout(width='200px',
margin='10px 0px'))

output = widgets.Output()

# Header

header = widgets.HTML(

    value="<h2 style='text-align: center;'>Patient Data Input</h2>",

    layout=widgets.Layout(margin='10px 0px')

)

# Submit button callback
```

```
def on_submit_clicked(b):
    with output:
        output.clear_output()
        age = age_widget.value
        sex = sex_widget.value
        bp = bp_widget.value
        cholesterol = cholesterol_widget.value
        na_to_k = na_to_k_widget.value
        new_patient_data = pd.DataFrame({
            'Age': [age],
            'Sex': [sex],
            'BP': [bp],
            'Cholesterol': [cholesterol],
            'Na_to_K': [na_to_k]
        })
        print("Submitted Data:")
        print(new_patient_data)
submit_button.on_click(on_submit_clicked)
# Arrange widgets in a vertical box
form = widgets.VBox([
    header,
    widgets.HBox([age_label, age_widget],
layout=widgets.Layout(justify_content='space-between', width='400px')),
    widgets.HBox([sex_label, sex_widget],
layout=widgets.Layout(justify_content='space-between', width='400px')),
    widgets.HBox([bp_label, bp_widget],
layout=widgets.Layout(justify_content='space-between', width='400px')),
    widgets.HBox([cholesterol_label, cholesterol_widget],
layout=widgets.Layout(justify_content='space-between', width='400px')),
```

```
    widgets.HBox([na_to_k_label, na_to_k_widget],
layout=widgets.Layout(justify_content='space-between', width='400px')),

    submit_button,

    output

], layout=widgets.Layout(align_items='center', width='50%',
margin='auto'))

# Display the form

display(form)
```

Output



Code

```
new_patient_data['Sex'] =
sex_encoding.fit_transform(new_patient_data['Sex'])

new_patient_data['BP'] =
BP_encoding.fit_transform(new_patient_data['BP'])

new_patient_data['Cholesterol'] =
Cholesterol_encoding.fit_transform(new_patient_data['Cholesterol'])

print("New Patient Data:")

print(new_patient_data)
```

Output

```
New Patient Data:
   Age  Sex  BP  Cholesterol  Na_to_K
0   30    0   0            0     15.0
```

Code

```
dt_prediction=dectree.predict(new_patient_data)

rf_prediction = ranforest.predict(new_patient_data)

print("\nPredictions for the New Patient:")

print("Decision Tree Prediction:", dt_prediction[0])

print("Random Forest Prediction:", rf_prediction[0])
```

Output

```
Predictions for the New Patient:
Decision Tree Prediction: drugY
Random Forest Prediction: drugY
```