

ML LAB ASSIGNMENT

SUPRATIM NAG -- CSE-AIML/22/057 -- GROUP-B

Q-5:Implementation of Decision Tree

(a) Use own dataset for the prediction using Decision Tree classifier. Split the dataset into training and test dataset in 80:20 ratio. Train the model on training dataset and use the test dataset for the prediction purpose.

```
In [2]: import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

```
In [3]: data = pd.read_csv(r"C:\Users\SUPRATIM NAG\OneDrive\Documents\ML\Personal_Datasets\Dataset.csv")
data.head(1)
```

```
Out[3]:
```

	Patient ID	Age	Blood Pressure	Cholesterol Levels	Heart Rate	BMI	Diagnosis	Treatment Plan	Recovery Status	Medication Type	Follow-up Requirement
0	101	65	130	250	72	28.0	Hypertension with high cholesterol.	Medication: Lisinopril (blood pressure), Stati...	Active Recovery	Lisinopril, Statins.	Quarterly.

```
In [4]: data.shape
```

```
Out[4]: (100, 11)
```

```
In [5]: input = data[['Age', 'Blood Pressure', 'Cholesterol Levels', 'Heart Rate', 'BMI', 'Diagnosis']]
input.head(1)
```

```
Out[5]:
```

	Age	Blood Pressure	Cholesterol Levels	Heart Rate	BMI	Diagnosis
0	65	130	250	72	28.0	Hypertension with high cholesterol.

```
In [6]: input['Diagnoses'] = input['Diagnosis'].apply(
lambda x: 1 if 'Hypertension' in x else 0
)
```

C:\Users\SUPRATIM NAG\AppData\Local\Temp\ipykernel_6740\1222192863.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
input['Diagnoses'] = input['Diagnosis'].apply(
```

```
In [7]: input.head(1)
```

```
Out[7]:
```

	Age	Blood Pressure	Cholesterol Levels	Heart Rate	BMI	Diagnosis	Diagnoses
0	65	130	250	72	28.0	Hypertension with high cholesterol.	1

```
In [8]: input.drop(axis=1, columns='Diagnosis', inplace=True)
```

C:\Users\SUPRATIM NAG\AppData\Local\Temp\ipykernel_6740\2096392200.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
input.drop(axis=1, columns='Diagnosis', inplace=True)
```

```
In [9]: input.describe()
```

```
Out[9]:
```

	Age	Blood Pressure	Cholesterol Levels	Heart Rate	BMI	Diagnoses
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	49.210000	126.100000	199.850000	79.380000	26.970000	0.310000
std	13.121264	15.018171	37.121524	7.947784	4.255015	0.464823
min	25.000000	95.000000	120.000000	60.000000	18.000000	0.000000
25%	39.750000	115.000000	177.500000	75.000000	24.000000	0.000000
50%	50.000000	125.000000	200.000000	80.000000	27.000000	0.000000
75%	60.000000	135.000000	226.250000	85.000000	30.000000	1.000000
max	75.000000	160.000000	300.000000	110.000000	40.000000	1.000000

```
In [10]: input.shape
```

```
Out[10]: (100, 6)
```

```
In [11]: features = input[['Blood Pressure', 'Age', 'Cholesterol Levels', 'Heart Rate', 'BMI']]
features.shape
```

```
Out[11]: (100, 5)
```

```
In [12]: target = input[['Diagnoses']]
target.shape
```

Out[12]: (100, 1)

In [13]: `X_train, X_test, y_train, y_test = train_test_split(features, target, test_size= 0.20)`

In [14]: `X_test.shape`

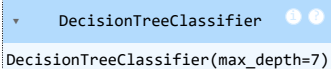
Out[14]: (20, 5)

In [15]: `model = DecisionTreeClassifier(max_depth=7)`

In [17]: `# model = DecisionTreeClassifier(criterion='gini', max_depth=7)`

In [19]: `# model = DecisionTreeClassifier(criterion='entropy', max_depth=7)`

In [16]: `model.fit(X_train, y_train)`

Out[16]: 

In [17]: `model.score(X_train, y_train)`

Out[17]: 0.975

In [18]: `model.score(X_test, y_test)`

Out[18]: 0.55

In [19]: `model.feature_importances_`

Out[19]: array([0.01581028, 0.32096868, 0.26878575, 0.21050097, 0.18393433])

In [20]: `print(model.predict([[125,80,200,70,28]]),"this is my Diagnosis status")`

[0] this is my Diagnosis status

c:\Users\SUPRATIM NAG\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
warnings.warn(

In [21]: `# dataset value`

`print(model.predict([[130, 65, 250, 72, 28]]),"this is my Diagnosis status")`

[1] this is my Diagnosis status

c:\Users\SUPRATIM NAG\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
warnings.warn(

In [22]: `X_test.head(5)`

Out[22]:

	Blood Pressure	Age	Cholesterol Levels	Heart Rate	BMI
74	125	35	190	80	27.0
39	150	70	240	90	32.0
88	135	42	230	90	33.0
93	110	30	170	80	21.0
47	125	55	200	70	26.0

In [23]: `y_test.head(5)`

Out[23]:

	Diagnoses
74	1
39	0
88	1
93	0
47	0