Pract5.ipynb Practical5.ipynb

Practical 5 > Practical5.ipynb > Prediction of Drug

+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Variables | Outline | ...

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[7] ✓ 0.8s

Loading of data

```
df = pd.read_csv("drugclassification.csv")
```

[8] ✓ 0.4s

Information of data

```
df.info()
```

[9] ✓ 0.1s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 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 
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

Pract5.ipynb Practical5.ipynb

Practical 5 > Practical5.ipynb > Prediction of Drug

+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Variables | Outline ...

Visualising dataframe

```
df.head()
```

[10] ✓ 0.6s

	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

Seperation of input and target data

```
inputs_val = df.drop('Drug',axis='columns')
target_val = df['Drug']
```

[11] ✓ 0.7s

```
Seperation of input and target data
```

```
inputs_val = df.drop('Drug',axis='columns')
target_val = df['Drug']
```

1] ✓ 0.7s

```
Applying label encoder
```

```
from sklearn.preprocessing import LabelEncoder
sex_up = LabelEncoder()
bp_up = LabelEncoder()
cho_up = LabelEncoder()
```

2] ✓ 0.6s

```
inputs_val['sex_up'] = sex_up.fit_transform(inputs_val['Sex'])
inputs_val['bp_up'] = bp_up.fit_transform(inputs_val['BP'])
inputs_val['cho_up'] = cho_up.fit_transform(inputs_val['Cholesterol'])
```

4] ✓ 0.7s

```
inputs_val
```

] ✓ 0.6s

	Age	Sex	BP	Cholesterol	Na_to_K	sex_up	bp_up	cho_up
0	23	F	HIGH	HIGH	25.355	0	0	0
1	47	M	LOW	HIGH	13.093	1	1	0
2	47	M	LOW	HIGH	10.114	1	1	0
3	28	F	NORMAL	HIGH	7.798	0	2	0
4	61	F	LOW	HIGH	18.043	0	1	0
...	...	...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	0	1	0
196	16	M	LOW	HIGH	12.006	1	1	0
197	52	M	NORMAL	HIGH	9.894	1	2	0
198	23	M	NORMAL	NORMAL	14.020	1	2	1
199	40	F	LOW	NORMAL	11.349	0	1	1

200 rows × 8 columns

```
inputs_n = inputs_val.drop(['Age','Sex','BP','Cholesterol'],axis='columns')
```

] ✓ 0.4s

```
inputs_n
```

✓ 0.1s

	Na_to_K	sex_up	bp_up	cho_up
0	25.355	0	0	0
1	13.093	1	1	0
2	10.114	1	1	0
3	7.798	0	2	0
4	18.043	0	1	0
...	...	...	...	...
195	11.567	0	1	0
196	12.006	1	1	0
197	9.894	1	2	0
198	14.020	1	2	1
199	11.349	0	1	1

200 rows × 4 columns

```
Formation of decision tree
```

```
from sklearn import tree
dt = tree.DecisionTreeClassifier()
```

[3] ✓ 0.4s

```
fitting the data into decision tree
```

```
dt.fit(inputs_n, target_val)
```

[4] ✓ 0.6s

```
DecisionTreeClassifier()
```

```
dt.score(inputs_n,target_val)
```

[5] ✓ 0.1s

```
1.0
```

```
Prediction of Drug
```

```
dt.predict([[25.355,0,0,0]])
```

[26] ✓ 0.3s

```
C:\Users\astha\AppData\Roaming\Python\Python310\site
names, but DecisionTreeClassifier was fitted with fe
  warnings.warn(

array(['DrugY'], dtype=object)
```

```
dt.predict([[12.56,0,1,2]])
```

[27] ✓ 0.4s

```
C:\Users\astha\AppData\Roaming\Python\Python310\site
names, but DecisionTreeClassifier was fitted with fe
  warnings.warn(

array(['drugX'], dtype=object)
```

```
tree.plot_tree(dt)
```

✓ 2.6s

Output exceeds the size limit. Open the full output data in a text editor

```
[Text(0.703125, 0.9615384615384616, 'X[0] <= 14.829\ngini = 0.694\nsamples = 200\nvalue = [91, 23, 16, 16, 54]'),
 Text(0.640625, 0.8846153846153846, 'X[2] <= 0.5\ngini = 0.667\nsamples = 109\nvalue = [0, 23, 16, 16, 54]'),
 Text(0.40625, 0.8076923076923077, 'X[0] <= 13.623\ngini = 0.484\nsamples = 39\nvalue = [0, 23, 16, 0, 0]'),
 Text(0.25, 0.7307692307692307, 'X[0] <= 8.316\ngini = 0.457\nsamples = 34\nvalue = [0, 22, 12, 0, 0]'),
 Text(0.1875, 0.6538461538461539, 'gini = 0.0\nsamples = 3\nvalue = [0, 3, 0, 0, 0]'),
 Text(0.3125, 0.6538461538461539, 'X[0] <= 10.24\ngini = 0.475\nsamples = 31\nvalue = [0, 19, 12, 0, 0]'),
 Text(0.1875, 0.5769230769230769, 'X[0] <= 9.67\ngini = 0.48\nsamples = 10\nvalue = [0, 4, 6, 0, 0]'),
 Text(0.125, 0.5, 'X[0] <= 8.661\ngini = 0.444\nsamples = 6\nvalue = [0, 4, 2, 0, 0]'),
 Text(0.0625, 0.4230769230769231, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1, 0, 0]'),
 Text(0.1875, 0.4230769230769231, 'X[3] <= 0.5\ngini = 0.32\nsamples = 5\nvalue = [0, 4, 1, 0, 0]'),
 Text(0.125, 0.34615384615384615, 'X[0] <= 9.615\ngini = 0.5\nsamples = 2\nvalue = [0, 1, 1, 0, 0]'),
 Text(0.0625, 0.2692307692307692, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1, 0, 0]'),
 Text(0.1875, 0.2692307692307692, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0, 0, 0]'),
 Text(0.25, 0.34615384615384615, 'gini = 0.0\nsamples = 3\nvalue = [0, 3, 0, 0, 0]'),
 Text(0.25, 0.5, 'gini = 0.0\nsamples = 4\nvalue = [0, 0, 4, 0, 0]'),
 Text(0.4375, 0.5769230769230769, 'X[0] <= 10.712\ngini = 0.408\nsamples = 21\nvalue = [0, 15, 6, 0, 0]'),
 Text(0.375, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0, 3, 0, 0, 0]'),
 Text(0.5, 0.5, 'X[0] <= 11.103\ngini = 0.444\nsamples = 18\nvalue = [0, 12, 6, 0, 0]'),
 Text(0.4375, 0.4230769230769231, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2, 0, 0]'),
 Text(0.5625, 0.4230769230769231, 'X[0] <= 11.333\ngini = 0.375\nsamples = 16\nvalue = [0, 12, 4, 0, 0]'),
 Text(0.5, 0.34615384615384615, 'gini = 0.0\nsamples = 4\nvalue = [0, 4, 0, 0, 0]'),
 Text(0.625, 0.34615384615384615, 'X[0] <= 11.607\ngini = 0.444\nsamples = 12\nvalue = [0, 8, 4, 0, 0]'),
 Text(0.5625, 0.2692307692307692, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2, 0, 0]'),
 Text(0.6875, 0.2692307692307692, 'X[0] <= 13.197\ngini = 0.32\nsamples = 10\nvalue = [0, 8, 2, 0, 0]'),
 Text(0.5625, 0.19230769230769232, 'X[0] <= 12.63\ngini = 0.219\nsamples = 8\nvalue = [0, 7, 1, 0, 0]'),
...
 Text(0.8125, 0.7307692307692307, 'X[3] <= 0.5\ngini = 0.498\nsamples = 34\nvalue = [0, 0, 0, 16, 18]'),
 Text(0.75, 0.6538461538461539, 'gini = 0.0\nsamples = 16\nvalue = [0, 0, 0, 16, 0]'),
 Text(0.875, 0.6538461538461539, 'gini = 0.0\nsamples = 18\nvalue = [0, 0, 0, 0, 18]'),
 Text(0.9375, 0.7307692307692307, 'gini = 0.0\nsamples = 36\nvalue = [0, 0, 0, 0, 36]'),
 Text(0.765625, 0.8846153846153846, 'gini = 0.0\nsamples = 91\nvalue = [91, 0, 0, 0, 0]')]
```