decision-tree

December 18, 2023

0.1 This is a project of drug prediction using decision tree algorithm

0.1.1 Importing neccessary libraries

[25]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[26]:
```
df = pd.read_csv("drug.csv")
```

[27]:
```
df.head(5)
```

[27]:

	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

0.1.2 Converting string values to numeric values for model fitting using LabelEncoder()

[28]:
```
from sklearn.preprocessing import LabelEncoder
```

[29]:
```
le = LabelEncoder()
```

[30]:
```
df.Sex = le.fit_transform(df.Sex)
df.BP = le.fit_transform(df.BP)
df.Cholesterol = le.fit_transform(df.Cholesterol)
df.Drug = le.fit_transform(df.Drug)
```

[31]:
```
df.head(5)
```

[31]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	0	0	25.355	4
1	47	1	1	0	13.093	2
2	47	1	1	0	10.114	2
3	28	0	2	0	7.798	3

```
4   61    0   1            0   18.043     4
```

0.1.3 Seperating data into input and target variables

[36]:
```
inputs = df.drop(["Drug"],axis ='columns')
inputs
```

[36]:
```
     Age  Sex  BP  Cholesterol  Na_to_K
0     23    0   0            0   25.355
1     47    1   1            0   13.093
2     47    1   1            0   10.114
3     28    0   2            0    7.798
4     61    0   1            0   18.043
..   …  …  ..          …       …
195   56    0   1            0   11.567
196   16    1   1            0   12.006
197   52    1   2            0    9.894
198   23    1   2            1   14.020
199   40    0   1            1   11.349

[200 rows x 5 columns]
```

[38]:
```
targets = df["Drug"]
targets
```

[38]:
```
0      4
1      2
2      2
3      3
4      4
      ..
195    2
196    2
197    3
198    3
199    3
Name: Drug, Length: 200, dtype: int32
```

0.1.4 Model fitting

[39]:
```
from sklearn import tree
```

[40]:
```
model = tree.DecisionTreeClassifier()
```

[41]:
```
model.fit(inputs,targets)
```

[41]:
```
DecisionTreeClassifier()
```

[42]:
```
model.score(inputs,targets)
```

[42]:
```
1.0
```

[43]:
```
model.predict([[23,0,0,0,25.355]])
```

```
C:\Users\luhar\anaconda3\lib\site-packages\sklearn\base.py:465: UserWarning: X
does not have valid feature names, but DecisionTreeClassifier was fitted with
feature names
  warnings.warn(
```

[43]:
```
array([4])
```

0.1.5 Here we got model score 1 wich is excellent and we predicted the drugY(4) for 25 year old female with high BP, high cholesterol and 25.355 unit amount of sodium_pottasium

0.1.6 Prediction by splitting data into testing and training

[44]:
```
from sklearn.model_selection import train_test_split
```

[48]:
```
xtrain, xtest, ytrain, ytest = train_test_split(inputs, targets, test_size=0.3)
```

[49]:
```
len(xtrain)
```

[49]:
```
140
```

[50]:
```
len(xtest)
```

[50]:
```
60
```

[51]:
```
model.fit(xtrain,ytrain)
```

[51]:
```
DecisionTreeClassifier()
```

[52]:
```
model.score(xtrain,ytrain)
```

[52]:
```
1.0
```

[54]:
```
model.predict(xtest)
```

[54]:
```
array([4, 4, 0, 3, 2, 0, 4, 1, 0, 2, 1, 0, 3, 4, 4, 2, 4, 3, 4, 4, 4, 4,
       4, 3, 3, 3, 3, 3, 3, 1, 0, 3, 3, 1, 2, 4, 0, 4, 3, 3, 4, 4, 4, 4,
       2, 2, 4, 0, 3, 4, 1, 3, 4, 2, 4, 0, 3, 4, 3, 2])
```

0.1.7 After splitting also we got the model score 1. We predicted the drug using our input tesing data and it matches 100% with target tesing data

[ ]: