

Experiment 8 Aim: Implementation of Decision tree using sklearn. Course Outcomes: CO4 and CO5

```
In [1]: import pandas as pd
        from sklearn.tree import DecisionTreeClassifier
        import matplotlib.pyplot as plt
```

```
In [3]: df = pd.read_csv("Exp-8 shows.csv")
```

```
In [4]: df
```

```
Out[4]:
```

	Age	Experience	Rank	Nationality	Go
0	36	10	9	UK	NO
1	42	12	4	USA	NO
2	23	4	6	N	NO
3	52	4	4	USA	NO
4	43	21	8	USA	YES
5	44	14	5	UK	NO
6	66	3	7	N	YES
7	35	14	9	UK	YES
8	52	13	7	N	YES
9	35	5	9	N	YES
10	24	3	5	USA	NO
11	18	3	7	UK	YES
12	45	9	9	UK	YES

```
In [5]: d = {'UK': 0, 'USA': 1, 'N': 2}
        df['Nationality'] = df['Nationality'].map(d)
        d = {'YES': 1, 'NO': 0}
        df['Go'] = df['Go'].map(d)
        print(df)
```

	Age	Experience	Rank	Nationality	Go
0	36	10	9	0	0
1	42	12	4	1	0
2	23	4	6	2	0
3	52	4	4	1	0
4	43	21	8	1	1
5	44	14	5	0	0
6	66	3	7	2	1
7	35	14	9	0	1
8	52	13	7	2	1
9	35	5	9	2	1
10	24	3	5	1	0
11	18	3	7	0	1
12	45	9	9	0	1

```
In [6]: features = ['Age', 'Experience', 'Rank', 'Nationality']

X = df[features]
y = df['Go']
```

```
In [7]: dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)
```

```
In [8]: print(dtree.predict([[40, 10, 7, 1]]))
print("[1] means 'GO'")
print("[0] means 'NO'")
```

```
[0]
[1] means 'GO'
[0] means 'NO'
```

```
C:\Users\admin\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
warnings.warn(
```

Part-2 Build a decision tree model for drug dataset About the dataset Imagine that you are a medical researcher compiling data for a study. You have collected data about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of 5 medications, Drug A, Drug B, Drug c, Drug x and y.

Part of your job is to build a model to find out which drug might be appropriate for a future patient with the same illness. The features of this dataset are Age, Sex, Blood Pressure, and the Cholesterol of the patients, and the target is the drug that each patient responded to.

It is a sample of multiclass classifier, and you can use the training part of the dataset to build a decision tree, and then use it to predict the class of a unknown patient, or to prescribe a drug to a new patient.

```
In [18]: df = pd.read_csv('Exp8_DT.csv')
df.head()
```

```
Out[18]:
```

	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

```
In [21]: from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
input = df.iloc[:, :-1]
Target = df.iloc[:, -1]
le_sex = LabelEncoder()
le_BP = LabelEncoder()
le_cholesterol = LabelEncoder()
input['Sex'] = le_sex.fit_transform(input['Sex'])
input['BP'] = le_BP.fit_transform(input['BP'])
input['Cholesterol'] = le_cholesterol.fit_transform(input['Cholesterol'])
input
```

Out[21]:

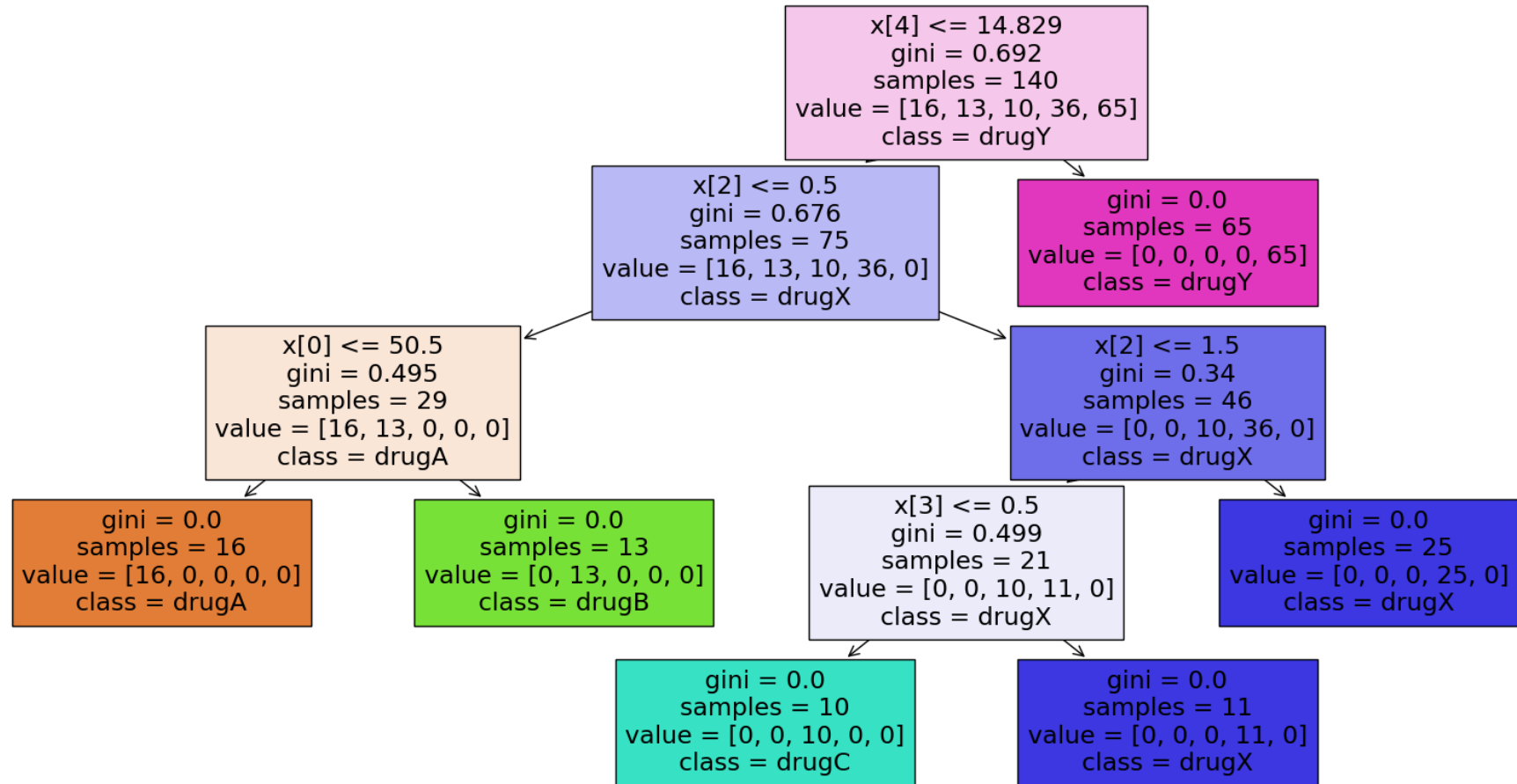
	Age	Sex	BP	Cholesterol	Na_to_K
0	23	0	0	0	25.355
1	47	1	1	0	13.093
2	47	1	1	0	10.114
3	28	0	2	0	7.798
4	61	0	1	0	18.043
...
195	56	0	1	0	11.567
196	16	1	1	0	12.006
197	52	1	2	0	9.894
198	23	1	2	1	14.020
199	40	0	1	1	11.349

200 rows × 5 columns

```
In [22]: X_train, X_test, y_train, y_test = train_test_split(input, Target, test_size=0.3, random_state=42)
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
accuracy = clf.score(X_test, y_test)
print("Model Accuracy:", round(accuracy*100,1) )
```

Model Accuracy: 100.0

```
In [24]: import numpy as np
from sklearn.tree import plot_tree
unique_classes = np.unique(y_train)
plt.figure(figsize=(20,10))
plot_tree(clf, class_names=unique_classes, filled=True)
plt.show()
```



```
In [25]: y_pred = clf.predict(X_test)
```

```
In [26]: from sklearn.metrics import classification_report, confusion_matrix, precision_score, recall_score, accuracy_score
```

```
In [28]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
drugA	1.00	1.00	1.00	7
drugB	1.00	1.00	1.00	3
drugC	1.00	1.00	1.00	6
drugX	1.00	1.00	1.00	18
drugY	1.00	1.00	1.00	26
accuracy			1.00	60
macro avg	1.00	1.00	1.00	60
weighted avg	1.00	1.00	1.00	60

```
In [29]: print(confusion_matrix(y_test, y_pred))
```

```
[[ 7  0  0  0  0]
 [ 0  3  0  0  0]
 [ 0  0  6  0  0]
 [ 0  0  0 18  0]
 [ 0  0  0  0 26]]
```

```
In [31]: print(precision_score(y_test, y_pred, average='micro'))
```

```
1.0
```

```
In [33]: print(recall_score(y_test, y_pred, average='macro'))
```

```
1.0
```

```
In [35]: print(accuracy_score(y_test, y_pred))
```

```
1.0
```

Learning Outcomes I Got to know about decision tree supervised learning algorithm for classification. Got to know about how to predict using decision tree Got to know how to draw a decision tree

Result/ Conclusion We have successfully trained the model and predicted the results.