```
import numpy as np
import pandas as pd
```

```
df=pd.read_csv("drug.csv")
df.head()
```

	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

Next steps: Generate code with df New interactive sheet

```
from sklearn.preprocessing import LabelEncoder

le_sex = LabelEncoder()
le_bp = LabelEncoder()
le_chol = LabelEncoder()
le_drug = LabelEncoder()

df['Sex'] = le_sex.fit_transform(df['Sex'])
df['BP'] = le_bp.fit_transform(df['BP'])
df['Cholesterol'] = le_chol.fit_transform(df['Cholesterol'])
df['Drug'] = le_drug.fit_transform(df['Drug'])

df.head()
```

	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

Next steps: Generate code with df New interactive sheet

```
X = df[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].values
y = df['Drug'].values
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(criterion='entropy', random_state=42)
model.fit(X_train, y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=42)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy of Decision Tree: {accuracy:.4f}\n")

cm = confusion_matrix(y_test, y_pred)
print("CONFUSION MATRIX (Text Based for B&W Print):\n")
print("CONFUSION MATRIX (Text-Based for B&W Print):\n")
unique_labels = sorted(set(y_test))
print("           Predicted")
print("          | " + " | ".join([f"Drug{le_drug.inverse_transform([lbl])[0][-1]}" for lbl in unique_labels]) + " |")
for i, row in enumerate(cm):
    print(f"Actual Drug{le_drug.inverse_transform([unique_labels[i]])[0][-1]} | " + " | ".join(f"{val:6d}" for val in row) +

print("\nCLASSIFICATION REPORT:\n")
print(classification_report(y_test, y_pred, target_names=[f"Drug{le_drug.inverse_transform([lbl])[0][-1]}" for lbl in unique
```

```
Accuracy of Decision Tree: 1.0000

CONFUSION MATRIX (Text-Based for B&W Print):

           Predicted
          | DrugA | DrugB | DrugC | DrugX | DrugY |
Actual DrugA |      6 |      0 |      0 |      0 |      0 |
Actual DrugB |      0 |      3 |      0 |      0 |      0 |
Actual DrugC |      0 |      0 |      5 |      0 |      0 |
Actual DrugX |      0 |      0 |      0 |     11 |      0 |
Actual DrugY |      0 |      0 |      0 |      0 |     15 |

CLASSIFICATION REPORT:

              precision    recall  f1-score   support

       DrugA       1.00      1.00      1.00         6
       DrugB       1.00      1.00      1.00         3
       DrugC       1.00      1.00      1.00         5
       DrugX       1.00      1.00      1.00        11
       DrugY       1.00      1.00      1.00        15

    accuracy                           1.00        40
   macro avg       1.00      1.00      1.00        40
weighted avg       1.00      1.00      1.00        40
```