In [5]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn import tree
```

In [6]:

```python
data=pd.read_csv(r"C:\Users\DHANANJAY KUMAR SING\Downloads\Decision_Tree_ Dataset
data
```

Out[6]:

	1	2	3	4	sum	Unnamed: 5
0	201	10018	250	3046	13515	yes
1	205	10016	395	3044	13660	yes
2	257	10129	109	3251	13746	yes
3	246	10064	324	3137	13771	yes
4	117	10115	496	3094	13822	yes
...	...	...	...	...	...	...
995	413	14914	523	4683	20533	No
996	359	14423	927	4838	20547	No
997	316	14872	613	4760	20561	No
998	305	14926	897	4572	20700	No
999	168	14798	834	4937	20737	No

1000 rows × 6 columns

In [16]:

```python
data.rename(columns={'1':'Initial payement','2':'Last payement','3':' Credit Scor
```

In [17]:
```
data
```

Out[17]:

	Initial payement	Last payement	Credit Score	House Number	sum	Result
0	201	10018	250	3046	13515	yes
1	205	10016	395	3044	13660	yes
2	257	10129	109	3251	13746	yes
3	246	10064	324	3137	13771	yes
4	117	10115	496	3094	13822	yes
...	...	...	...	...	...	...
995	413	14914	523	4683	20533	No
996	359	14423	927	4838	20547	No
997	316	14872	613	4760	20561	No
998	305	14926	897	4572	20700	No
999	168	14798	834	4937	20737	No

1000 rows × 6 columns

In [18]:
```
data.describe().transpose()
```

Out[18]:

	count	mean	std	min	25%	50%	75%	max
Initial payement	1000.0	294.343	115.815390	100.0	195.00	289.5	398.00	500.0
Last payement	1000.0	12465.884	1440.156170	10005.0	11201.50	12450.0	13678.25	14996.0
Credit Score	1000.0	528.042	259.796059	100.0	302.00	516.5	753.50	997.0
House Number	1000.0	4027.011	565.164179	3003.0	3545.00	4041.5	4507.00	5000.0
sum	1000.0	17315.280	1589.943875	13515.0	16073.75	17383.0	18595.25	20737.0

In [19]:
```
data.isnull().sum()
```

Out[19]:
```
Initial payement    0
Last payement       0
 Credit Score       0
House Number        0
sum                 0
Result              0
dtype: int64
```

In [20]:
```
data.shape
```

Out[20]:
```
(1000, 6)
```

In [21]:
```
data.columns
```

Out[21]:
```
Index(['Initial payement', 'Last payement', ' Credit Score', 'House Number',
       'sum', 'Result'],
      dtype='object')
```

In [22]:
```
len(data)
```

Out[22]:
```
1000
```

In [23]:
```
data.head()
```

Out[23]:

	Initial payement	Last payement	Credit Score	House Number	sum	Result
0	201	10018	250	3046	13515	yes
1	205	10016	395	3044	13660	yes
2	257	10129	109	3251	13746	yes
3	246	10064	324	3137	13771	yes
4	117	10115	496	3094	13822	yes

In [24]:
```
X=data.values[:, 1:5];
Y=data.values[:, 5]
```

In [25]:
```

X_train,X_test,y_train,y_test=train_test_split(X, Y,test_size=0.3,random_state=1
```

In [ ]:
```
#celll added
```

In [26]:
```

clf_entropy=DecisionTreeClassifier(criterion="entropy",
                 random_state=100,max_depth=3)
clf_entropy.fit(X_train,y_train)

```

Out[26]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=100)
```

In [27]:

```
#function to make prediction
y_pred_en=clf_entropy.predict(X_test)
print(y_pred_en)
```

```
['yes' 'yes' 'No' 'yes' 'No' 'yes' 'No' 'No' 'No' 'No' 'No' 'No' 'yes'
 'No' 'No' 'yes' 'yes' 'No' 'yes' 'No' 'No' 'yes' 'No' 'yes' 'yes' 'No'
 'No' 'yes' 'No' 'No' 'No' 'yes' 'yes' 'yes' 'yes' 'No' 'No' 'No' 'yes'
 'yes' 'yes' 'yes' 'yes' 'No' 'No' 'yes' 'No' 'yes' 'No' 'No' 'yes' 'No'
 'yes' 'yes' 'yes' 'yes' 'No' 'No' 'No' 'yes' 'yes' 'No' 'No' 'yes' 'No'
 'yes' 'yes' 'yes' 'No' 'yes' 'No' 'No' 'No' 'yes' 'No' 'yes' 'yes' 'No'
 'yes' 'No' 'No' 'No' 'No' 'yes' 'No' 'yes' 'No' 'No' 'yes' 'No' 'yes'
 'yes' 'No' 'yes' 'yes' 'No' 'yes' 'yes' 'yes' 'yes' 'No' 'No' 'yes' 'No'
 'No' 'yes' 'yes' 'yes' 'No' 'yes' 'yes' 'yes' 'yes' 'No' 'No' 'yes' 'yes'
 'yes' 'No' 'yes' 'No' 'No' 'yes' 'yes' 'No' 'No' 'yes' 'yes' 'No' 'No'
 'yes' 'yes' 'No' 'yes' 'yes' 'yes' 'No' 'yes' 'yes' 'No' 'No' 'yes' 'yes'
 'yes' 'No' 'yes' 'yes' 'yes' 'yes' 'No' 'yes' 'No' 'No' 'No' 'No' 'yes'
 'No' 'No' 'No' 'yes' 'No' 'No' 'yes' 'yes' 'No' 'yes' 'No' 'yes' 'No'
 'No' 'yes' 'yes' 'No' 'yes' 'No' 'yes' 'yes' 'No' 'No' 'No' 'No' 'No'
 'yes' 'No' 'No' 'yes' 'yes' 'yes' 'yes' 'yes' 'No' 'yes' 'No' 'yes' 'No'
 'No' 'No' 'yes' 'yes' 'No' 'No' 'yes' 'yes' 'No' 'No' 'No' 'yes' 'yes'
 'No' 'No' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'No'
 'yes' 'yes' 'No' 'No' 'yes' 'No' 'yes' 'yes' 'No' 'No' 'yes' 'No' 'yes'
 'yes' 'yes' 'yes' 'No' 'No' 'No' 'No' 'No' 'yes' 'No' 'yes' 'yes' 'No'
```

In [32]:

```
print("Accuracy is",accuracy_score(y_test,y_pred_en)*100)
```

```
Accuracy is 100.0
```

In [29]:

```
print('Matrix:', confusion_matrix(y_test, y_pred_en, labels=['yes', 'No']))
```

```
Matrix: [[153   0]
 [  0 147]]
```