In [27]:
```
ASSIGNMENT 7
```

In [35]:
```
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.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score, confusion_matrix, classification_r
eport
```

In [5]:
```
df=pd.read_csv("PlayTennis.csv")
display(df.head())
```

	Outlook	Temperature	Humidity	Wind	Play Tennis
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes

In [6]:
```
Outlook = df["Outlook"].str.get_dummies(sep=" ")
```

In [7]:
```
Temperature = df["Temperature"].str.get_dummies(sep=" ")
```

In [8]:
```
Humidity = df["Humidity"].str.get_dummies(sep=" ")
```

In [9]:
```
Wind = df["Wind"].str.get_dummies(sep=" ")
```

In [10]:
```
play_Tennis = df["Play Tennis"].str.get_dummies(sep=" ")
```

In [11]:
```
df.drop(['Outlook','Temperature','Humidity','Wind','Play Tennis'],axis=1,inpla
ce=True);
```

In [12]:
```
print(Outlook)
```

```
    Overcast  Rain  Sunny
0          0     0      1
1          0     0      1
2          1     0      0
3          0     1      0
4          0     1      0
5          0     1      0
6          1     0      0
7          0     0      1
8          0     0      1
9          0     1      0
10         0     0      1
11         1     0      0
12         1     0      0
13         0     1      0
```

In [13]:
```
print(Outlook)
print(Temperature)
print(Humidity)
print(Wind)
print(play_Tennis)
```

	Overcast	Rain	Sunny
0	0	0	1
1	0	0	1
2	1	0	0
3	0	1	0
4	0	1	0
5	0	1	0
6	1	0	0
7	0	0	1
8	0	0	1
9	0	1	0
10	0	0	1
11	1	0	0
12	1	0	0
13	0	1	0

	Cool	Hot	Mild
0	0	1	0
1	0	1	0
2	0	1	0
3	0	0	1
4	1	0	0
5	1	0	0
6	1	0	0
7	0	0	1
8	1	0	0
9	0	0	1
10	0	0	1
11	0	0	1
12	0	1	0
13	0	0	1

	High	Normal
0	1	0
1	1	0
2	1	0
3	1	0
4	0	1
5	0	1
6	0	1
7	1	0
8	0	1
9	0	1
10	0	1
11	1	0
12	0	1
13	1	0

	Strong	Weak
0	0	1
1	1	0
2	0	1
3	0	1
4	0	1
5	1	0
6	1	0
7	0	1
8	0	1
9	0	1
10	1	0

```
11       1     0
12       0     1
13       1     0
    No  Yes
0    1    0
1    1    0
2    0    1
3    0    1
4    0    1
5    1    0
6    0    1
7    1    0
8    0    1
9    0    1
10   0    1
11   0    1
12   0    1
13   1    0
```

In [14]:
```
df = pd.concat([Outlook, Temperature, Humidity, Wind, play_Tennis], axis=1)
```

In [15]:
```
df
```

Out[15]:

	Overcast	Rain	Sunny	Cool	Hot	Mild	High	Normal	Strong	Weak	No	Yes
0	0	0	1	0	1	0	1	0	0	1	1	0
1	0	0	1	0	1	0	1	0	1	0	1	0
2	1	0	0	0	1	0	1	0	0	1	0	1
3	0	1	0	0	0	1	1	0	0	1	0	1
4	0	1	0	1	0	0	0	1	0	1	0	1
5	0	1	0	1	0	0	0	1	1	0	1	0
6	1	0	0	1	0	0	0	1	1	0	0	1
7	0	0	1	0	0	1	1	0	0	1	1	0
8	0	0	1	1	0	0	0	1	0	1	0	1
9	0	1	0	0	0	1	0	1	0	1	0	1
10	0	0	1	0	0	1	0	1	1	0	0	1
11	1	0	0	0	0	1	1	0	1	0	0	1
12	1	0	0	0	1	0	0	1	0	1	0	1
13	0	1	0	0	0	1	1	0	1	0	1	0

In [16]:
```
x = df.drop(['Yes', 'No'], axis=1)
```

In [17]:
```
y=df['Yes']
```

In [30]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, strat
ify=y)
```

In [31]:
```
dt = DecisionTreeClassifier(criterion='entropy')
```

In [32]:
```
dt.fit(x_train,y_train)
```

Out[32]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=Non
e,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

In [33]:
```
y_pred=dt.predict(x_test)
```

In [34]:
```
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print(accuracy_score(y_test, y_pred))
```

```
[[2 0]
 [2 1]]
             precision    recall  f1-score   support

          0       0.50      1.00      0.67         2
          1       1.00      0.33      0.50         3

avg / total       0.80      0.60      0.57         5

0.6
```