In [1]:

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

In [2]:

```
df=pd.read_csv('play_tennis.csv')
```

In [3]:

```
df
```

Out[3]:

	day	outlook	temp	humidity	wind	play
0	D1	Sunny	Hot	High	Weak	No
1	D2	Sunny	Hot	High	Strong	No
2	D3	Overcast	Hot	High	Weak	Yes
3	D4	Rain	Mild	High	Weak	Yes
4	D5	Rain	Cool	Normal	Weak	Yes
5	D6	Rain	Cool	Normal	Strong	No
6	D7	Overcast	Cool	Normal	Strong	Yes
7	D8	Sunny	Mild	High	Weak	No
8	D9	Sunny	Cool	Normal	Weak	Yes
9	D10	Rain	Mild	Normal	Weak	Yes
10	D11	Sunny	Mild	Normal	Strong	Yes
11	D12	Overcast	Mild	High	Strong	Yes
12	D13	Overcast	Hot	Normal	Weak	Yes
13	D14	Rain	Mild	High	Strong	No

In [4]:

```
df.shape
```

Out[4]:

```
(14, 6)
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   day       14 non-null     object
 1   outlook   14 non-null     object
 2   temp      14 non-null     object
 3   humidity  14 non-null     object
 4   wind      14 non-null     object
 5   play      14 non-null     object
dtypes: object(6)
memory usage: 804.0+ bytes
```

In [9]:

```
df.describe()
```

Out[9]:

	day	outlook	temp	humidity	wind	play
count	14	14	14	14	14	14
unique	14	3	3	2	2	2
top	D1	Sunny	Mild	High	Weak	Yes
freq	1	5	6	7	8	9

In [7]:

```
df.isnull()
```

Out[7]:

	day	outlook	temp	humidity	wind	play
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
5	False	False	False	False	False	False
6	False	False	False	False	False	False
7	False	False	False	False	False	False
8	False	False	False	False	False	False
9	False	False	False	False	False	False
10	False	False	False	False	False	False
11	False	False	False	False	False	False
12	False	False	False	False	False	False
13	False	False	False	False	False	False

In [10]:

```
df.head()
```

Out[10]:

	day	outlook	temp	humidity	wind	play
0	D1	Sunny	Hot	High	Weak	No
1	D2	Sunny	Hot	High	Strong	No
2	D3	Overcast	Hot	High	Weak	Yes
3	D4	Rain	Mild	High	Weak	Yes
4	D5	Rain	Cool	Normal	Weak	Yes

In [12]:

```
df['play'].value_counts()
```

Out[12]:

```
play
Yes    9
No     5
Name: count, dtype: int64
```

In [13]:

```
df['play'].unique()
```

Out[13]:

```
array(['No', 'Yes'], dtype=object)
```

Approach 1 Label Encoding

In [14]:

```
df_encoded=pd.DataFrame()
df_encoded
```

Out[14]: —

In [18]:

```
from sklearn.preprocessing import LabelEncoder
le= LabelEncoder()
```

In [19]:

```
for col in df.columns:
    df_encoded[col]=le.fit_transform(df[col])
```

In [20]:

```
df_encoded
```

Out[20]:

	day	outlook	temp	humidity	wind	play
0	0	2	1	0	1	0
1	6	2	1	0	0	0
2	7	0	1	0	1	1
3	8	1	2	0	1	1
4	9	1	0	1	1	1
5	10	1	0	1	0	0
6	11	0	0	1	0	1
7	12	2	2	0	1	0
8	13	2	0	1	1	1
9	1	1	2	1	1	1
10	2	2	2	1	0	1
11	3	0	2	0	0	1
12	4	0	1	1	1	1
13	5	1	2	0	0	0

In [24]:

```
X=df_encoded.iloc[:,0:5]
X
```

Out[24]:

	day	outlook	temp	humidity	wind
0	0	2	1	0	1
1	6	2	1	0	0
2	7	0	1	0	1
3	8	1	2	0	1
4	9	1	0	1	1
5	10	1	0	1	0
6	11	0	0	1	0
7	12	2	2	0	1
8	13	2	0	1	1
9	1	1	2	1	1
10	2	2	2	1	0
11	3	0	2	0	0
12	4	0	1	1	1
13	5	1	2	0	0

In [25]:

```
y=df_encoded.iloc[:,-1]
y
```

Out[25]:
```
0     0
1     0
2     1
3     1
4     1
5     0
6     1
7     0
8     1
9     1
10    1
11    1
12    1
13    0
Name: play, dtype: int32
```

In [26]:
```
from sklearn.linear_model import LogisticRegression
```

In [27]:
```
clf=LogisticRegression()
```

In [28]:
```
clf.fit(X,y)
```

Out[28]:
```
▾ LogisticRegression ⓘ ?
LogisticRegression()
```

In [29]:
```
clf.score(X,y)
```

Out[29]:
```
0.8571428571428571
```

In [30]:
```
12/14 # 2 miss classifications
```

Out[30]:
```
0.8571428571428571
```

Approach 2 One-Hot Encoding

In [31]:
```
from sklearn.preprocessing import OneHotEncoder
```

In [46]:
```
oe=OneHotEncoder(sparse_output=False)
```

In [47]:
```
column_list=['outlook','temp','humidity','wind']
```

In [48]:
```
encoded=oe.fit_transform(df[column_list])
```

In [49]:
```
encoded
```

Out[49]:
```
array([[0., 0., 1., 0., 1., 0., 1., 0., 0., 1.],
       [0., 0., 1., 0., 1., 0., 1., 0., 1., 0.],
       [1., 0., 0., 0., 1., 0., 1., 0., 0., 1.],
       [0., 1., 0., 0., 0., 1., 1., 0., 0., 1.],
       [0., 1., 0., 1., 0., 0., 0., 1., 0., 1.],
       [0., 1., 0., 1., 0., 0., 0., 1., 1., 0.],
       [1., 0., 0., 1., 0., 0., 0., 1., 1., 0.],
       [0., 0., 1., 0., 0., 1., 1., 0., 0., 1.],
       [0., 0., 1., 1., 0., 0., 0., 1., 0., 1.],
       [0., 1., 0., 0., 0., 1., 0., 1., 0., 1.],
       [0., 0., 1., 0., 0., 1., 0., 1., 1., 0.],
       [1., 0., 0., 0., 0., 1., 1., 0., 1., 0.],
       [1., 0., 0., 0., 1., 0., 0., 1., 0., 1.],
       [0., 1., 0., 0., 0., 1., 1., 0., 1., 0.]])
```

In [54]:
```
X=pd.DataFrame(encoded,columns=oe.get_feature_names_out(column_list))         #type casted array-to-Data
X
```

Out[54]:

	outlook_Overcast	outlook_Rain	outlook_Sunny	temp_Cool	temp_Hot	temp_Mild	humidity_High	humidity_N
0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	
1	0.0	0.0	1.0	0.0	1.0	0.0	1.0	
2	1.0	0.0	0.0	0.0	1.0	0.0	1.0	
3	0.0	1.0	0.0	0.0	0.0	1.0	1.0	
4	0.0	1.0	0.0	1.0	0.0	0.0	0.0	
5	0.0	1.0	0.0	1.0	0.0	0.0	0.0	
6	1.0	0.0	0.0	1.0	0.0	0.0	0.0	
7	0.0	0.0	1.0	0.0	0.0	1.0	1.0	
8	0.0	0.0	1.0	1.0	0.0	0.0	0.0	
9	0.0	1.0	0.0	0.0	0.0	1.0	0.0	
10	0.0	0.0	1.0	0.0	0.0	1.0	0.0	
11	1.0	0.0	0.0	0.0	0.0	1.0	1.0	
12	1.0	0.0	0.0	0.0	1.0	0.0	0.0	
13	0.0	1.0	0.0	0.0	0.0	1.0	1.0	

In [55]:

```
clf2=LogisticRegression()
clf2.fit(X,y)
```

Out[55]:

```
▾ LogisticRegression
LogisticRegression()
```

In [56]:

```
clf2.score(X,y)
```

Out[56]:

```
0.8571428571428571
```

In [57]:

```
from sklearn.metrics import confusion_matrix
```

In [58]:

```
cm=confusion_matrix(y,clf2.predict(X))
```

In [59]:

```
cm
```

Out[59]:

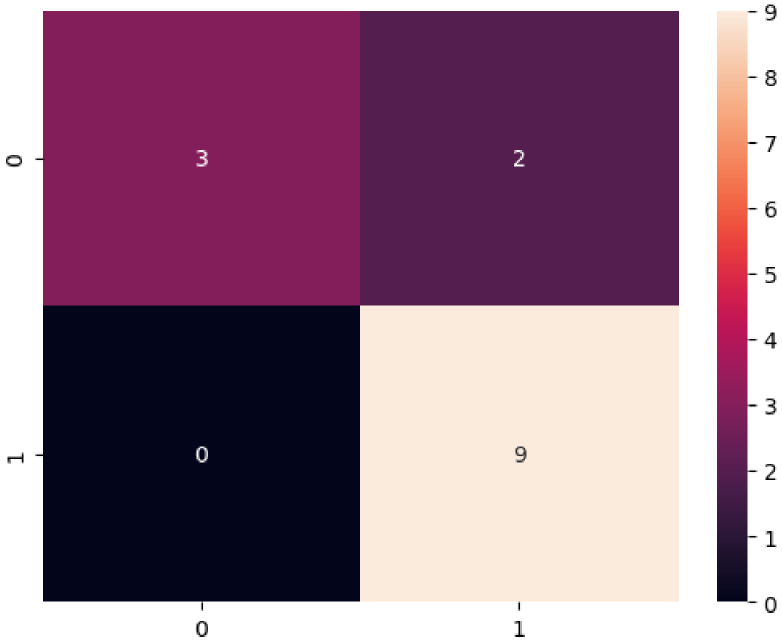
```
array([[3, 2],
       [0, 9]], dtype=int64)
```

In [60]:

```
import seaborn as sns
sns.heatmap(cm,annot=True)
```

Out[60]:

```
<Axes: >
```



```
In [ ]:
```