In [48]:
```
import pandas as pd
```

In [49]:
```
df = pd.read_csv("play_tennis_train.csv")
```

In [50]:
```
df
```

Out[50]:

	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 [51]:
```
inputs=df.drop('play',axis='columns')
inputs
```

Out[51]:

	day	outlook	temp	humidity	wind
0	D1	Sunny	Hot	High	Weak
1	D2	Sunny	Hot	High	Strong
2	D3	Overcast	Hot	High	Weak
3	D4	Rain	Mild	High	Weak
4	D5	Rain	Cool	Normal	Weak
5	D6	Rain	Cool	Normal	Strong
6	D7	Overcast	Cool	Normal	Strong
7	D8	Sunny	Mild	High	Weak
8	D9	Sunny	Cool	Normal	Weak
9	D10	Rain	Mild	Normal	Weak
10	D11	Sunny	Mild	Normal	Strong
11	D12	Overcast	Mild	High	Strong
12	D13	Overcast	Hot	Normal	Weak
13	D14	Rain	Mild	High	Strong

In [52]:

```
from sklearn.preprocessing import LabelEncoder
le_outlook = LabelEncoder()
le_temp = LabelEncoder()
le_humidity = LabelEncoder()
le_wind = LabelEncoder()
```

In [53]:

```
inputs['outlook_n']=le_outlook.fit_transform(inputs['outlook'])
inputs['temp_n']=le_temp.fit_transform(inputs['temp'])
inputs['humidity_n']=le_humidity.fit_transform(inputs['humidity'])
inputs['wind_n']=le_wind.fit_transform(inputs['wind'])
```

In [54]:

```
inputs
```

Out[54]:

	day	outlook	temp	humidity	wind	outlook_n	temp_n	humidity_n	wind_n
0	D1	Sunny	Hot	High	Weak	2	1	0	1
1	D2	Sunny	Hot	High	Strong	2	1	0	0
2	D3	Overcast	Hot	High	Weak	0	1	0	1
3	D4	Rain	Mild	High	Weak	1	2	0	1
4	D5	Rain	Cool	Normal	Weak	1	0	1	1
5	D6	Rain	Cool	Normal	Strong	1	0	1	0
6	D7	Overcast	Cool	Normal	Strong	0	0	1	0
7	D8	Sunny	Mild	High	Weak	2	2	0	1
8	D9	Sunny	Cool	Normal	Weak	2	0	1	1
9	D10	Rain	Mild	Normal	Weak	1	2	1	1
10	D11	Sunny	Mild	Normal	Strong	2	2	1	0
11	D12	Overcast	Mild	High	Strong	0	2	0	0
12	D13	Overcast	Hot	Normal	Weak	0	1	1	1
13	D14	Rain	Mild	High	Strong	1	2	0	0

In [55]:

```
inputs_n=inputs.drop(['day','outlook','temp','humidity','wind'],axis='columns')
```

In [56]:

```
inputs_n
```

Out[56]:

	outlook_n	temp_n	humidity_n	wind_n
0	2	1	0	1
1	2	1	0	0
2	0	1	0	1
3	1	2	0	1
4	1	0	1	1
5	1	0	1	0
6	0	0	1	0
7	2	2	0	1
8	2	0	1	1
9	1	2	1	1
10	2	2	1	0
11	0	2	0	0
12	0	1	1	1
13	1	2	0	0

In [57]:

```
target=df['play']
target
```

Out[57]:

```
0      No
1      No
2     Yes
3     Yes
4     Yes
5      No
6     Yes
7      No
8     Yes
9     Yes
10    Yes
11    Yes
12    Yes
13     No
Name: play, dtype: object
```

In [58]:

```
from sklearn import tree
model=tree.DecisionTreeClassifier(criterion='entropy',random_state=0)
```

In [61]:

```
model.fit(inputs_n, target)
```

Out[61]:

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

In [62]:

```
model.score(inputs_n,target)
```

Out[62]:

```
1.0
```

In [63]:

```
model.predict([[2,1,0,1]])
```

```
C:\Users\hp\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not
have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

Out[63]:

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

In [64]:

```
model.predict([[2,0,0,0]])
```

```
C:\Users\hp\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not
have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

Out[64]:

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

In [75]:

```
model.predict([[0,1,0,1]])
```

```
C:\Users\hp\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not
have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

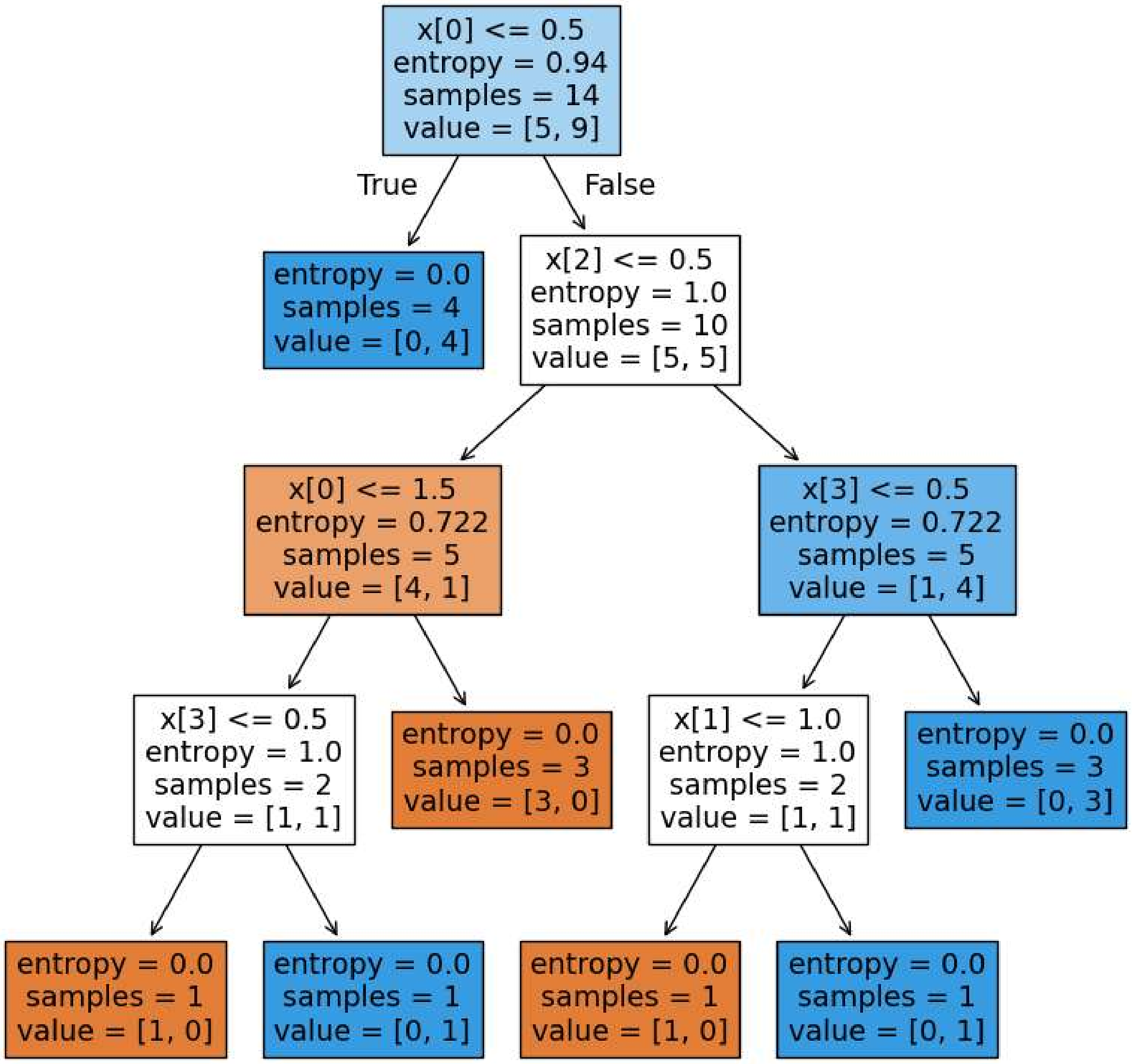
Out[75]:

```
array(['Yes'], dtype=object)
```

In [77]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
tree.plot_tree(model,filled=True)
#tree.plot_tree(model)
```

```
Out[77]: [Text(0.4444444444444444, 0.9, 'x[0] <= 0.5\nentropy = 0.94\nsamples = 14\nvalue =
         [5, 9]'),
          Text(0.3333333333333333, 0.7, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
          Text(0.38888888888888884, 0.8, 'True  '),
          Text(0.5555555555555556, 0.7, 'x[2] <= 0.5\nentropy = 1.0\nsamples = 10\nvalue =
         [5, 5]'),
          Text(0.5, 0.8, '  False'),
          Text(0.3333333333333333, 0.5, 'x[0] <= 1.5\nentropy = 0.722\nsamples = 5\nvalue =
         [4, 1]'),
          Text(0.2222222222222222, 0.3, 'x[3] <= 0.5\nentropy = 1.0\nsamples = 2\nvalue =
         [1, 1]'),
          Text(0.1111111111111111, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
          Text(0.3333333333333333, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
          Text(0.4444444444444444, 0.3, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
          Text(0.7777777777777778, 0.5, 'x[3] <= 0.5\nentropy = 0.722\nsamples = 5\nvalue =
         [1, 4]'),
          Text(0.6666666666666666, 0.3, 'x[1] <= 1.0\nentropy = 1.0\nsamples = 2\nvalue =
         [1, 1]'),
          Text(0.5555555555555556, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
          Text(0.7777777777777778, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
          Text(0.8888888888888888, 0.3, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]')]
```

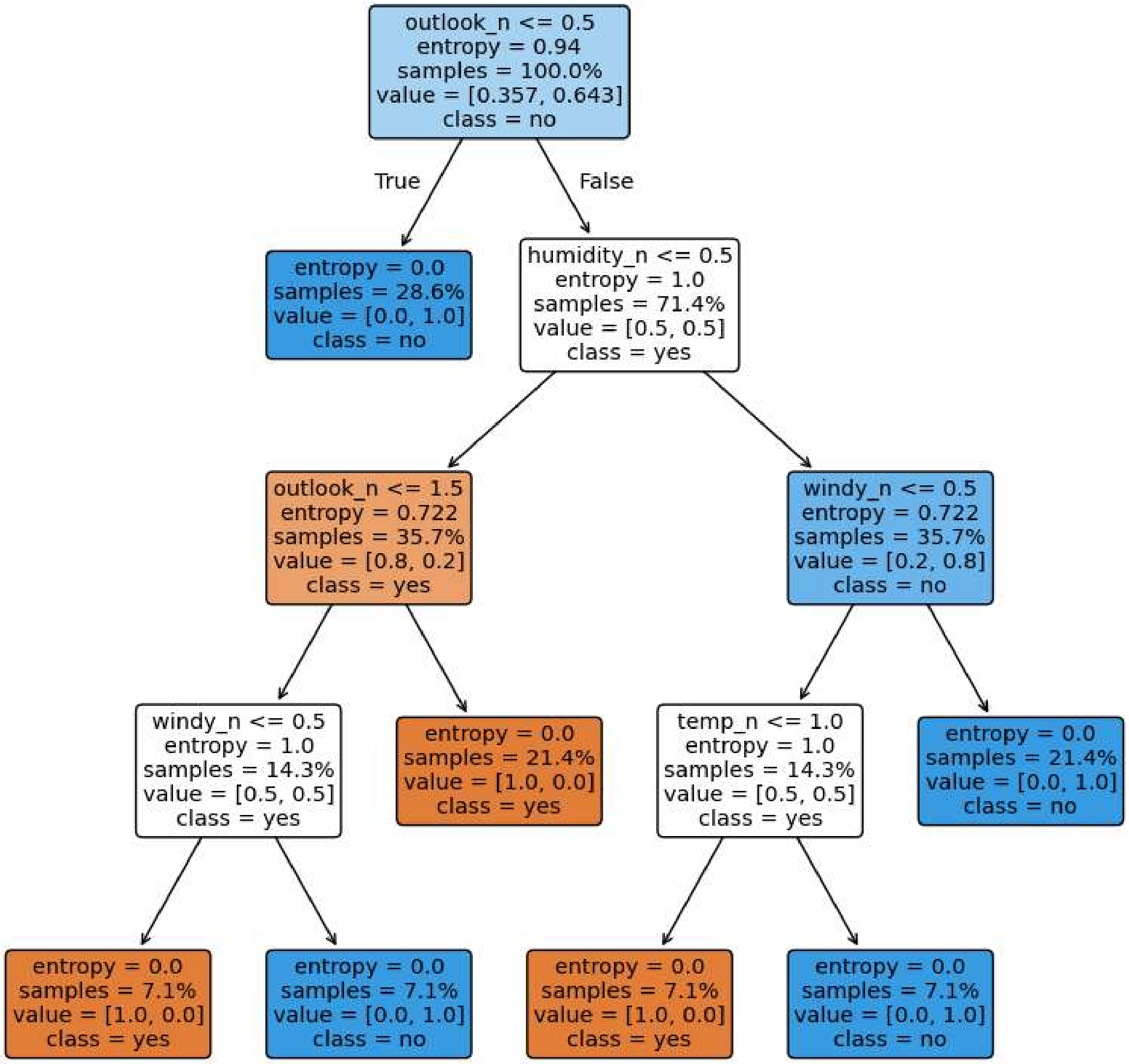


```
In [80]: print(tree.export_text(model))
```

```
|--- feature_0 <= 0.50
|   |--- class: Yes
|--- feature_0 >  0.50
|   |--- feature_2 <= 0.50
|   |   |--- feature_0 <= 1.50
|   |   |   |--- feature_3 <= 0.50
|   |   |   |   |--- class: No
|   |   |   |--- feature_3 >  0.50
|   |   |   |   |--- class: Yes
|   |   |--- feature_0 >  1.50
|   |   |   |--- class: No
|   |--- feature_2 >  0.50
|   |   |--- feature_3 <= 0.50
|   |   |   |--- feature_1 <= 1.00
|   |   |   |   |--- class: No
|   |   |   |--- feature_1 >  1.00
|   |   |   |   |--- class: Yes
|   |   |--- feature_3 >  0.50
|   |   |   |--- class: Yes
```

In [94]:

```
plt.figure(figsize=(10,10))
feature_cols = ['outlook_n','temp_n','humidity_n','windy_n']
tree.plot_tree(model,
               class_names=['yes','no'],
               filled=True,
               rounded=True,
               proportion=True,
               feature_names=feature_cols, max_depth=8)
plt.show()
```



In [96]:
```
from sklearn.model_selection import train_test_split
```

In [97]:
```
x_train, x_test, y_train, y_test= train_test_split(inputs_n,target,test_size=0.2,ra
```

In [100…
```
len(x_train)
```

Out[100…
```
11
```

In [101…
```
len(x_test)
```

Out[101…
```
3
```

In [102…
```
from sklearn import metrics
```

In [103…
```
#create decision tree classifier
clf = tree.DecisionTreeClassifier(criterion='entropy',random_state=0)
clf = clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
```

In [111…

```
#training score
clf.score(x_train,y_train)
```

Out[111…

```
1.0
```

In [113…

```
print("Accuracy:",metrics.accuracy_score(y_test,y_pred))
```

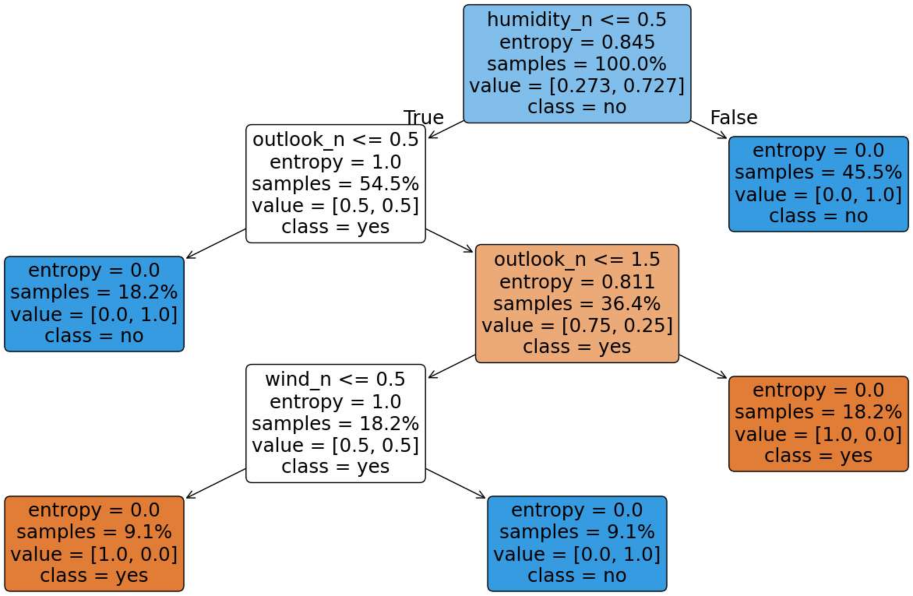
```
Accuracy: 0.6666666666666666
```

In [115…

```
feature_cols = ['outlook_n','temp_n','humidity_n', 'wind_n']
```

In [117…

```
plt.figure(figsize=(20,10))
tree.plot_tree(clf,
               class_names=['yes','no'],
               filled=True,
               rounded=True,
               proportion=True,
               feature_names=feature_cols, max_depth=8)
plt.show()
```



In [ ]:

In [ ]: