```
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
from sklearn import metrics
```

```
df=pd.read_csv("/content/Play Tennis.csv")
value=['Outlook','Temprature','Humidity','Wind']
df
```

	Day	Outlook	Temprature	Humidity	Wind	Play_Tennis
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

```
len(df)
```

```
14
```

```
df.shape
```

```
(14, 6)
```

```
df.head(9)
```

	Day	Outlook	Temprature	Humidity	Wind	Play_Tennis
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

```
df.tail(8)
```

	Day	Outlook	Temprature	Humidity	Wind	Play_Tennis

```
df.describe()
```

	Day	Outlook	Temprature	Humidity	Wind	Play_Tennis
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

```
from sklearn import preprocessing
string_to_int= preprocessing.LabelEncoder()                    #encode your data
df=df.apply(string_to_int.fit_transform) #fit and transform it
df
```

	Day	Outlook	Temprature	Humidity	Wind	Play_Tennis
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

```
#To divide our data into attribute set and Label:
feature_cols = ['Outlook','Temprature','Humidity','Wind']
X = df[feature_cols ]                                #contains the attribute
y = df.Play_Tennis                                   #contains the label
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

```
# perform training
from sklearn.tree import DecisionTreeClassifier                              # import the classifier
classifier =DecisionTreeClassifier(criterion="entropy", random_state=100)     # create a classifier object
classifier.fit(X_train, y_train)                                               # fit the classifier with X and Y data or
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=100)
```

```
#Predict the response for test dataset
y_pred= classifier.predict(X_test)

type(X_test)
```

```
pandas.core.frame.DataFrame
```

```
data_1 = {'state' : ['VA', 'VA', 'VA', 'MD', 'MD'],
          'year' : [2012, 2013, 2014, 2014, 2015],
          'pop' : [5.0, 5.1, 5.2, 4.0, 4.1]}
df_1 = DataFrame(data_1)
df_1
```

	state	year	pop
0	VA	2012	5.0
1	VA	2013	5.1
2	VA	2014	5.2
3	MD	2014	4.0

```
data_2 = {'Outlook' : ['2'], 'Temprature' : ['1'], 'Humidity' : ['0'], 'Wind' : ['1']}
df_2 = DataFrame(data_2)
df_2

y_pred2= classifier.predict(df_2)
y_pred2
```

```
array([0])
```

```
# Model Accuracy, how often is the classifier correct?
from sklearn.metrics import accuracy_score
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.2
```

```
data_p=pd.DataFrame({'Actual':y_test, 'Predicted':y_pred})
data_p
```

	Actual	Predicted
11	1	0
10	1	0
8	1	0
6	1	0
9	1	1

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

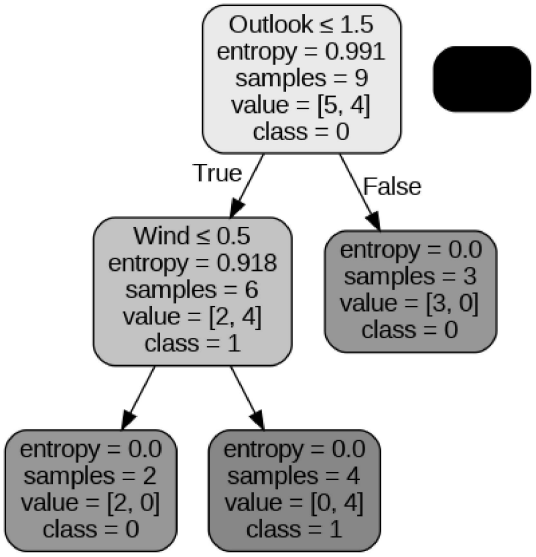
```
[[0 0]
 [4 1]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       1.00      0.20      0.33         5

    accuracy                           0.20         5
   macro avg       0.50      0.10      0.17         5
weighted avg       1.00      0.20      0.33         5

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill
  _warn_prf(average, modifier, msg_start, len(result))
```

```
from sklearn.tree import export_graphviz
from IPython.display import Image
import pydotplus
import io
dot_data = io.StringIO()
export_graphviz(classifier, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names=value, class_names=['0', '1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('Play_Tennis.png')
Image(graph.create_png())
```


Outlook ≤ 1.5
entropy = 0.991
samples = 9
value = [5, 4]
class = 0
True
False
Wind ≤ 0.5
entropy = 0.918
samples = 6
value = [2, 4]
class = 1
entropy = 0.0
samples = 3
value = [3, 0]
class = 0
entropy = 0.0
samples = 2
value = [2, 0]
class = 0
entropy = 0.0
samples = 4
value = [0, 4]
class = 1