des

April 4, 2024

1 ANANT KUMAR TIWARI

$2 \quad 211112230$

2.0.1 importing libraries

```
[1]: import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
```

3 Decision Tree for Play Tennis

```
[2]: col_names = ['outlook', 'temp', 'humidity', 'windy', 'play']
data = pd.read_csv("PlayTennis.csv", skiprows=1, header=None, names=col_names)
```

4 Encoding the column

```
[3]: Le = LabelEncoder()

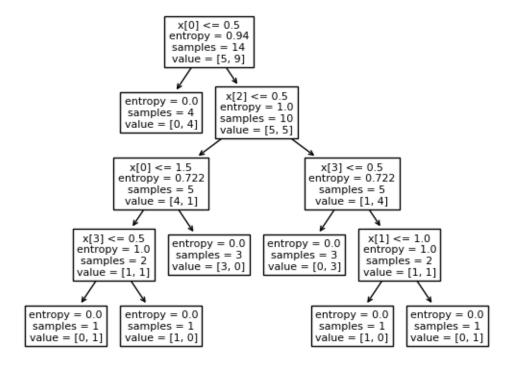
data['outlook'] = Le.fit_transform(data['outlook'])
data['temp'] = Le.fit_transform(data['temp'])
data['humidity'] = Le.fit_transform(data['humidity'])
data['windy'] = Le.fit_transform(data['windy'])
data['play'] = Le.fit_transform(data['play'])
```

```
[4]: X= data.iloc[:,:-1]
y= data.iloc[:,-1]
```

5 Building and Training the model

5.0.1 using entropy

```
[5]: from sklearn import tree
                                                               clf = tree.DecisionTreeClassifier(criterion = 'entropy')
                                                               clf = clf.fit(X, y)
[6]: tree.plot_tree(clf)
[6]: [Text(0.4, 0.9, 'x[0] \le 0.5 \le 0.94 \le 14 \le 14 \le [5, 9]')
                                                                           Text(0.3, 0.7, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
                                                                           Text(0.5, 0.7, 'x[2] \le 0.5 \le 1.0 \le
                                                                           Text(0.3, 0.5, 'x[0] \le 1.5 \cdot 0.722 \cdot
                                                                           Text(0.2, 0.3, 'x[3] \le 0.5 \cdot 1.0 \cdot
                                                                           Text(0.1, 0.1, 'entropy = 0.0 \setminus samples = 1 \setminus value = [0, 1]'),
                                                                           Text(0.3, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
                                                                           Text(0.4, 0.3, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
                                                                           Text(0.7, 0.5, 'x[3] \le 0.5 \cdot 0.722 \cdot
                                                                           Text(0.6, 0.3, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
                                                                           Text(0.8, 0.3, 'x[1] \le 1.0 \le 1.0 \le 2 \le 2 \le [1, 1]')
                                                                           Text(0.7, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
                                                                           Text(0.9, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]')
```



```
[7]: y_pred = clf.predict(X)
   [8]: y_pred == y
   [8]: 0
                                            True
                      1
                                            True
                      2
                                            True
                      3
                                            True
                      4
                                            True
                      5
                                            True
                      6
                                            True
                     7
                                           True
                      8
                                            True
                      9
                                            True
                      10
                                           True
                      11
                                            True
                      12
                                            True
                      13
                                            True
                      Name: play, dtype: bool
                              calculating the accuracy
                   6.0.1 using entropy
   [9]: from sklearn.metrics import accuracy_score
                      accuracy_score(y_pred, y)
   [9]: 1.0
                                 Using Gini index
[10]: clf2 = tree.DecisionTreeClassifier(criterion = 'gini')
                      clf2 = clf2.fit(X, y)
[11]: tree.plot_tree(clf2)
[11]: [Text(0.4, 0.9, 'x[0] \le 0.5 \text{ ngini} = 0.459 \text{ nsamples} = 14 \text{ nvalue} = [5, 9]'),
                         Text(0.3, 0.7, 'gini = 0.0 \setminus samples = 4 \setminus value = [0, 4]'),
                         Text(0.5, 0.7, 'x[2] \le 0.5 \le 0.5 \le 10 \le 10 \le 10 \le 10
                         Text(0.3, 0.5, 'x[0] \le 1.5 \text{ in } = 0.32 \text{ nsamples} = 5 \text{ nvalue} = [4, 1]'),
                         Text(0.2, 0.3, 'x[3] \le 0.5 = 0.5 = 2 = [1, 1]'),
                         Text(0.1, 0.1, 'gini = 0.0 \setminus samples = 1 \setminus value = [0, 1]'),
                         Text(0.3, 0.1, 'gini = 0.0 \setminus samples = 1 \setminus value = [1, 0]'),
                         Text(0.4, 0.3, 'gini = 0.0 \setminus samples = 3 \setminus value = [3, 0]'),
                         Text(0.7, 0.5, 'x[3] \le 0.5 \le 0.5 \le 0.3 \le 0.32 \le 0.3 \le 0.32 \le 0.3 \le 0.3
```

```
Text(0.6, 0.3, 'gini = 0.0 \setminus samples = 3 \setminus gini = [0, 3]'),
Text(0.8, 0.3, 'x[0] \le 1.5 \le 0.5 \le 2 \le [1, 1]'),
Text(0.7, 0.1, 'gini = 0.0 \setminus samples = 1 \setminus value = [1, 0]'),
Text(0.9, 0.1, 'gini = 0.0 \setminus samples = 1 \setminus value = [0, 1]')]
                                  x[0] <= 0.5
                                 gini = 0.459
                                 samples = 14
                                 value = [5, 9]
                                         x[2] \le 0.5
                           gini = 0.0
                                          gini = 0.5
                          samples = 4
                                        samples = 10
                         value = [0, 4]
                                        value = [5, 5]
                          x[0] \le 1.5
                                                        x[3] <= 0.5
                          gini = 0.32
                                                        gini = 0.32
                          samples = 5
                                                        samples = 5
                         value = [4, 1]
                                                       value = [1, 4]
                  x[3] <= 0.5
                                                               x[0] <= 1.5
                                                 gini = 0.0
                                  gini = 0.0
                   gini = 0.5
                                                                gini = 0.5
                                 samples = 3
                                                samples = 3
                  samples = 2
                                                               samples = 2
                                 value = [3, 0] value = [0, 3]
                  value = [1, 1]
                                                              value = [1, 1]
            gini = 0.0
                           gini = 0.0
                                                         gini = 0.0
                                                                        gini = 0.0
                                                                      samples = 1
           samples = 1
                          samples = 1
                                                        samples = 1
          value = [0, 1] value = [1, 0]
                                                       value = [1, 0]
                                                                      value = [0, 1]
```

```
[12]: y_pred = clf2.predict(X)
      y==y_pred
[12]: 0
             True
             True
      1
      2
             True
      3
             True
      4
             True
      5
             True
      6
             True
      7
             True
      8
             True
      9
             True
             True
      10
      11
             True
      12
             True
      13
             True
      Name: play, dtype: bool
```

```
[13]: from sklearn.metrics import accuracy_score accuracy_score(y_pred, y)
```

[13]: 1.0