In [1]:
```
import pandas as pd
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [2]:
```
pwd
```

Out[2]:
```
'C:\\Users\\Hadi\\Desktop\\data science analystics\\week7 machin learning\\Week-7-MachineLearning\\weather'
```

In [4]:
```
data = pd.read_csv('C:\\Users\\Hadi\\Desktop\\data science analystics\\week7 machin learning\\Week-7-MachineLearning\\weather\\daily_weather.csv')
```

In [5]:
```
data.columns
```

Out[5]:
```
Index(['number', 'air_pressure_9am', 'air_temp_9am', 'avg_wind_direction_9am',
       'avg_wind_speed_9am', 'max_wind_direction_9am', 'max_wind_speed_9am',
       'rain_accumulation_9am', 'rain_duration_9am', 'relative_humidity_9am',
       'relative_humidity_3pm'],
      dtype='object')
```

In [7]:
```
data[data.isnull().any(axis=1)]
```

Out[7]:

	number	air_pressure_9am	air_temp_9am	avg_wind_direction_9am	avg_wind_speed_9am	max_wind_direction_9am	max
16	16	917.899000	NaN	169.200000	2.192201	196.800000	
111	111	915.290000	58.820000	182.600000	15.613841	189.000000	
177	177	915.900000	NaN	183.300000	4.719943	189.900000	
262	262	923.596607	58.380598	47.737753	10.636273	67.145463	
277	277	920.480000	62.600000	194.400000	2.751436	NaN	
334	334	916.230000	75.740000	149.100000	2.751436	187.500000	
358	358	917.440000	58.514000	55.100000	10.021491	NaN	
361	361	920.444946	65.801645	49.822346	21.520177	61.898944	
381	381	918.480000	66.542000	90.900000	3.467257	89.400000	
409	409	NaN	67.853833	65.880616	4.328594	78.570823	
517	517	920.570000	53.600000	100.100000	4.697574	NaN	
519	519	919.250000	55.670000	176.400000	6.666081	188.200000	
546	546	NaN	42.746000	251.100000	12.929513	274.400000	
620	620	921.200000	56.786000	192.300000	9.551734	201.400000	
625	625	912.400000	50.774000	171.600000	NaN	181.400000	
656	656	920.830000	66.344000	NaN	15.457255	189.400000	
670	670	910.920000	48.362000	156.500000	NaN	177.500000	
672	672	922.448945	72.863773	NaN	3.682370	214.196160	
705	705	911.900000	59.072000	199.800000	1.279056	239.500000	
731	731	922.970166	51.391847	33.810942	NaN	59.290099	
737	737	917.895130	76.804690	104.771020	1.632705	97.178763	
788	788	917.923442	73.349717	42.101739	4.132698	64.284969	
840	840	918.042757	NaN	181.774042	0.964376	185.618601	
848	848	915.250000	37.562000	246.500000	11.587349	258.700000	
861	861	919.065408	NaN	172.303728	2.839600	193.058141	
869	869	NaN	45.104000	259.000000	3.265932	275.000000	
998	998	914.140000	71.240000	NaN	1.722444	232.900000	
1031	1031	922.669195	NaN	47.946284	7.969686	65.770066	
1035	1035	919.670000	77.576000	171.800000	6.554234	191.000000	
1063	1063	917.300195	65.790001	NaN	1.879553	222.496226	
1066	1066	919.564869	73.726732	69.704694	3.551777	102.573516	

In [8]:
```
del data['number']
```

In [10]:
```
before_rows = data.shape[0]
print(before_rows)
```

```
1095
```

In [11]:
```
data = data.dropna()
```

In [12]:
```
after_rows = data.shape[0]
print(after_rows)
```

```
1064
```

In [13]:
```
before_rows - after_rows
```

Out[13]:
```
31
```

Convert to a Classification Task

In [16]:
```
clean_data = data.copy()
clean_data['high_humidity_label'] = (clean_data['relative_humidity_3pm'] > 24.99)*1
print(clean_data['high_humidity_label'])
```

```
0       1
1       0
2       0
3       0
4       1
5       1
6       0
7       1
8       0
9       1
10      1
11      1
12      1
13      1
14      0
15      0
17      0
18      1
19      0
20      0
21      1
22      0
23      1
24      0
25      1
26      1
27      1
28      1
29      1
30      1
       ..
1064    1
1065    1
1067    1
1068    1
1069    1
1070    1
1071    0
1072    0
1073    1
1074    1
1075    0
1076    0
1077    1
1078    0
1079    1
1080    0
1081    0
1082    1
1083    1
1084    1
1085    1
1086    1
1087    1
1088    1
1089    1
1090    1
1091    1
1092    1
1093    1
1094    0
Name: high_humidity_label, Length: 1064, dtype: int32
```

In [18]:
```
y=clean_data[['high_humidity_label']].copy()
y
```

Out[18]:

	high_humidity_label
0	1
1	0
2	0
3	0
4	1
5	1
6	0
7	1
8	0
9	1
10	1
11	1
12	1
13	1
14	0
15	0
17	0
18	1
19	0
20	0
21	1
22	0
23	1
24	0
25	1
26	1
27	1
28	1
29	1
30	1
…	…
1064	1
1065	1
1067	1
1068	1
1069	1
1070	1
1071	0
1072	0
1073	1
1074	1
1075	0
1076	0
1077	1
1078	0
1079	1
1080	0
1081	0
1082	1
1083	1
1084	1
1085	1
1086	1
1087	1
1088	1
1089	1
1090	1
1091	1
1092	1
1093	1
1094	0

1064 rows × 1 columns

In [19]:
```
clean_data['relative_humidity_3pm'].head()
```

Out[19]:
```
0    36.160000
1    19.426597
2    14.469999
3    12.742547
4    76.740000
Name: relative_humidity_3pm, dtype: float64
```

In [20]:
```
y.head()
```

Out[20]:

	high_humidity_label
0	1
1	0
2	0
3	0
4	1

In [21]:
```
morning_features = ['air_pressure_9am','air_temp_9am','avg_wind_direction_9am','avg_wind_speed_9am',
                    'max_wind_direction_9am','max_wind_speed_9am','rain_accumulation_9am',
                    'rain_duration_9am']
```

In [22]:
```
X = clean_data[morning_features].copy() # for naother type of data base we shoud used deep = true.
```

In [23]:
```
X.columns
```

Out[23]:
```
Index(['air_pressure_9am', 'air_temp_9am', 'avg_wind_direction_9am',
       'avg_wind_speed_9am', 'max_wind_direction_9am', 'max_wind_speed_9am',
       'rain_accumulation_9am', 'rain_duration_9am'],
      dtype='object')
```

In [24]:
```
y.columns
```

Out[24]:
```
Index(['high_humidity_label'], dtype='object')
```

Perform Test and Train split

In [25]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=324)
```

In [31]:
```
type(X_train)
type(X_test)
type(y_train)
type(y_test)
X_train.head()
y_train.describe()
```

Out[31]:

	high_humidity_label
count	712.000000
mean	0.494382
std	0.500320
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

In [32]:
```
humidity_classifier = DecisionTreeClassifier(max_leaf_nodes=10, random_state=0)
humidity_classifier.fit(X_train, y_train)
```

Out[32]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=10,
                       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=0,
                       splitter='best')
```

In [33]:
```
type(humidity_classifier)
```

Out[33]:
```
sklearn.tree.tree.DecisionTreeClassifier
```

In [34]:
```
predictions = humidity_classifier.predict(X_test)
```

In [35]:
```
predictions[:10]
```

Out[35]:
```
array([0, 0, 1, 1, 1, 1, 0, 0, 0, 1])
```

In [36]:
```
y_test['high_humidity_label'][:10]
```

Out[36]:
```
456    0
845    0
693    1
259    1
723    1
224    1
300    1
442    0
585    1
1057   1
Name: high_humidity_label, dtype: int32
```

In [37]:
```
accuracy_score(y_true = y_test, y_pred = predictions)
```

Out[37]:
```
0.8153409090909091
```