

```
In [12]: import numpy as np
import pandas as pd
from sklearn import tree
from sklearn.metrics import confusion_matrix, accuracy_score, classification_
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
```

```
In [3]: Data=pd.read_csv("C:/Users/SAGNIK SAMANTA/OneDrive/Desktop/Datasets/Person_
print(Data)
```

	Age	Experience	Rank	Nationality	Go
0	36	10	9	UK	NO
1	42	12	4	USA	NO
2	23	4	6	N	NO
3	52	4	4	USA	NO
4	43	21	8	USA	YES
5	44	14	5	UK	NO
6	66	3	7	N	YES
7	35	14	9	UK	YES
8	52	13	7	N	YES
9	35	5	9	N	YES
10	24	3	5	USA	NO
11	18	3	7	UK	YES
12	45	9	9	UK	YES

To make a Decision Tree, All data has to be numerical.

We have to convert the non numerical columns 'Nationality' and 'Go' into numerical values.

Pandas has a map() method that takes a dictionary with information on how to convert the values.

```
{'UK': 0, 'USA': 1, 'N': 2}
```

Means convert the values 'UK' to 0, 'USA' to 1, and 'N' to 2.

```
In [4]: d={"UK":0,"USA":1,"N":2}
Data["Nationality"]=Data["Nationality"].map(d)
d={"YES":1,"NO":0}
Data["Go"]=Data["Go"].map(d)
print(Data)
```

	Age	Experience	Rank	Nationality	Go
0	36	10	9	0	0
1	42	12	4	1	0
2	23	4	6	2	0
3	52	4	4	1	0
4	43	21	8	1	1
5	44	14	5	0	0
6	66	3	7	2	1
7	35	14	9	0	1
8	52	13	7	2	1
9	35	5	9	2	1
10	24	3	5	1	0
11	18	3	7	0	1
12	45	9	9	0	1

Then we have to separate the feature columns from the target column.

The feature columns are the columns that we try to predict from, and the target column is the column with the values we try to predict.

```
In [7]: features=["Age","Experience","Rank","Nationality"]
X=Data[features]
y=Data["Go"]
```

```
In [8]: print(X)
```

	Age	Experience	Rank	Nationality
0	36	10	9	0
1	42	12	4	1
2	23	4	6	2
3	52	4	4	1
4	43	21	8	1
5	44	14	5	0
6	66	3	7	2
7	35	14	9	0
8	52	13	7	2
9	35	5	9	2
10	24	3	5	1
11	18	3	7	0
12	45	9	9	0

```
In [9]: print(y)
```

```
0      0
1      0
2      0
3      0
4      1
5      0
6      1
7      1
8      1
9      1
10     0
11     1
12     1
Name: Go, dtype: int64
```

```
In [10]: dtree=DecisionTreeClassifier()
dtree.fit(X,y)
```

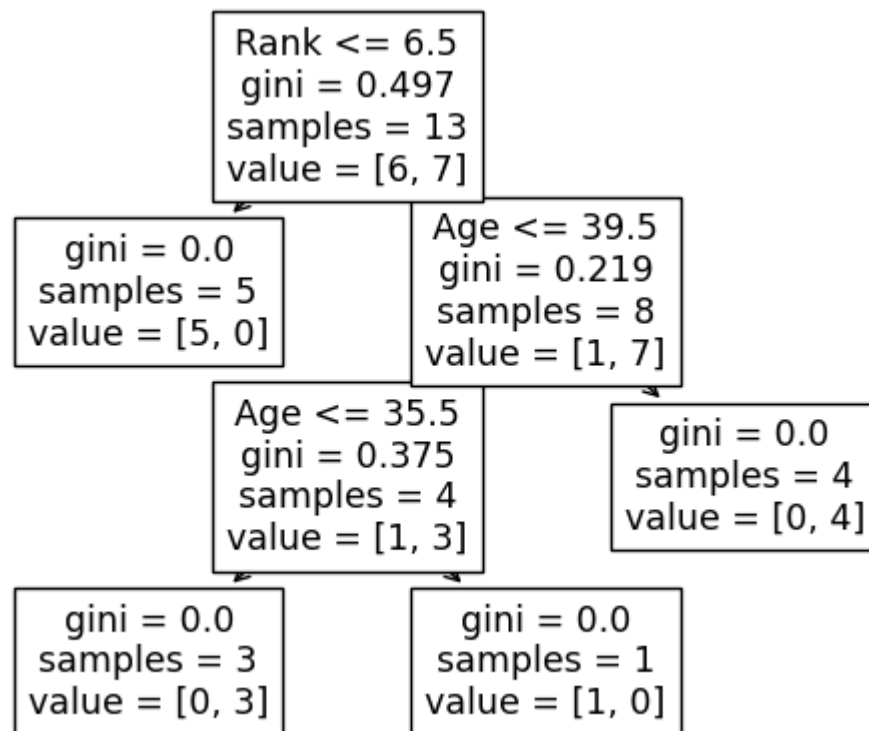
Out[10]: DecisionTreeClassifier()

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [14]: tree.plot_tree(dtree, feature_names=features)
```

```
Out[14]: [Text(0.4, 0.875, 'Rank <= 6.5\ngini = 0.497\nsamples = 13\nvalue = [6, 7]'),  
Text(0.2, 0.625, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),  
Text(0.6, 0.625, 'Age <= 39.5\ngini = 0.219\nsamples = 8\nvalue = [1, 7]'),  
Text(0.4, 0.375, 'Age <= 35.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),  
Text(0.2, 0.125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),  
Text(0.6, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.8, 0.375, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]')]
```



```
In [ ]:
```