In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:
```
healthData = pd.read_csv('Health.csv')
print(healthData)
```

```
  Ethnicity  Height (CM)  Weight (Kg) Will survive till 70
0     White        186.0         90.0                  Yes
1   African        185.0         98.0                   No
2     White        175.0         80.0                   No
3     White        180.0         88.0                  Yes
4     Asian        178.0          NaN                   No
5     Asian        172.0         72.0                  Yes
6   African        178.0         75.0                   No
7     White          NaN         89.0                  Yes
8   African        186.0         90.0                  Yes
```

In [3]:
```
X = healthData.iloc[:, :-1].values
y = healthData.iloc[:,-1].values
```

In [7]:
```
X
```

Out[7]:
```
array([['White', 186.0, 90.0],
       ['African', 185.0, 98.0],
       ['White', 175.0, 80.0],
       ['White', 180.0, 88.0],
       ['Asian', 178.0, nan],
       ['Asian', 172.0, 72.0],
       ['African', 178.0, 75.0],
       ['White', nan, 89.0],
       ['African', 186.0, 90.0]], dtype=object)
```

In [8]:
```
y
```

Out[8]:
```
array(['Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes'],
      dtype=object)
```

In [10]:
```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer.fit(X[:, 1:3])
X[:, 1:3] = imputer.transform(X[:, 1:3])
print(X)
```

```
[['White' 186.0 90.0]
 ['African' 185.0 98.0]
 ['White' 175.0 80.0]
 ['White' 180.0 88.0]
 ['Asian' 178.0 85.25]
 ['Asian' 172.0 72.0]
 ['African' 178.0 75.0]
 ['White' 180.0 89.0]
 ['African' 186.0 90.0]]
```

In [11]:
```
from sklearn.preprocessing import LabelEncoder
X_labelencoder = LabelEncoder()
X[:, 0] = X_labelencoder.fit_transform(X[:, 0])
print(X)
```

```
[[2 186.0 90.0]
 [0 185.0 98.0]
 [2 175.0 80.0]
 [2 180.0 88.0]
 [1 178.0 85.25]
 [1 172.0 72.0]
 [0 178.0 75.0]
 [2 180.0 89.0]
 [0 186.0 90.0]]
```

In [12]:

```
y_labelencoder = LabelEncoder()
y = y_labelencoder.fit_transform(y)
print(y)
```

```
[1 0 0 1 0 1 0 1 1]
```

In [13]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='
X = np.array(ct.fit_transform(X))
print(X)
```

```
[[0.0 0.0 1.0 186.0 90.0]
 [1.0 0.0 0.0 185.0 98.0]
 [0.0 0.0 1.0 175.0 80.0]
 [0.0 0.0 1.0 180.0 88.0]
 [0.0 1.0 0.0 178.0 85.25]
 [0.0 1.0 0.0 172.0 72.0]
 [1.0 0.0 0.0 178.0 75.0]
 [0.0 0.0 1.0 180.0 89.0]
 [1.0 0.0 0.0 186.0 90.0]]
```

In [17]:

```
from sklearn.preprocessing import StandardScaler
independent_scalar = StandardScaler()
X = independent_scalar.fit_transform(X)
print(X)
```

```
[[-0.70710678 -0.53452248  1.11803399  1.29232469  0.61464681]
 [ 1.41421356 -0.53452248 -0.89442719  1.07693724  1.64984143]
 [-0.70710678 -0.53452248  1.11803399 -1.07693724 -0.67934647]
 [-0.70710678 -0.53452248  1.11803399  0.          0.35584815]
 [-0.70710678  1.87082869 -0.89442719 -0.4307749   0.        ]
 [-0.70710678  1.87082869 -0.89442719 -1.72309958 -1.71454109]
 [ 1.41421356 -0.53452248 -0.89442719 -0.4307749  -1.32634311]
 [-0.70710678 -0.53452248  1.11803399  0.          0.48524748]
 [ 1.41421356 -0.53452248 -0.89442719  1.29232469  0.61464681]]
```

In [18]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat
print(X_train, X_test, y_train, y_test, sep='\n')
```

```
[[ 1.41421356 -0.53452248 -0.89442719 -0.4307749  -1.32634311]
 [-0.70710678 -0.53452248  1.11803399  0.          0.48524748]
 [ 1.41421356 -0.53452248 -0.89442719  1.07693724  1.64984143]
 [-0.70710678 -0.53452248  1.11803399  1.29232469  0.61464681]
 [-0.70710678  1.87082869 -0.89442719 -0.4307749   0.        ]
 [-0.70710678 -0.53452248  1.11803399  0.          0.35584815]
 [-0.70710678  1.87082869 -0.89442719 -1.72309958 -1.71454109]]
[[ 1.41421356 -0.53452248 -0.89442719  1.29232469  0.61464681]
 [-0.70710678 -0.53452248  1.11803399 -1.07693724 -0.67934647]]
```

```
[0 1 0 1 0 1 1]
[1 0]
```

In [ ]: