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

In [2]:
```
df = pd.read_csv("mldata.csv")
df.head()
```

Out[2]:

	age	weight	gender	likeness	height
0	27	76.0	Male	Biryani	170.688
1	41	70.0	Male	Biryani	165
2	29	80.0	Male	Biryani	171
3	27	102.0	Male	Biryani	173
4	29	67.0	Male	Biryani	164

The decision tree acquires knowledge in the form of a tree, which can also be rewritten as a set of discrete rules to make it easier to understand. The main advantage of the decision tree classifier is its ability to using different feature subsets and decision rules at different stages of classification.

In [3]:
```
#Step Starting with dummy values and changing object to numeric value

df['gender'] = df['gender'].replace("Male", 1)
df["gender"] = df["gender"].replace("Female", 0)
df["height"] = pd.to_numeric(df["height"], errors='coerce')
```

In [4]:
```
#Dropping null value because it was created by the coerec argument in to_nume
df = df.dropna()
df.isnull().sum()
```

Out[4]:
```
age         0
weight      0
gender      0
likeness    0
height      0
dtype: int64
```

In [5]:
```
#Input or features and output or
x = df[["weight", "gender", "age", "height"]]
y = df["likeness"]
y.tail()
```

Out[5]:
```
240     Pakora
241    Biryani
242    Biryani
243    Biryani
244     Samosa
Name: likeness, dtype: object
```

In [6]:
```
from sklearn.model_selection import train_test_split
```

In [7]:
```
X_train, X_test, y_train, y_test = train_test_split(x,y, train_size=0.8, rand
```

In [8]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [9]:
```
dt =  DecisionTreeClassifier().fit(X_train, y_train)
dt
```

Out[9]:
```
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 [10]:
```
model = dt.predict(X_test)
model
```

Out[10]:
```
array(['Biryani', 'Biryani', 'Biryani', 'Biryani', 'Samosa', 'Pakora',
       'Pakora', 'Samosa', 'Biryani', 'Biryani', 'Samosa', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Pakora',
       'Biryani', 'Samosa', 'Samosa', 'Pakora', 'Samosa', 'Samosa',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Pakora',
       'Biryani', 'Samosa', 'Biryani', 'Samosa', 'Biryani', 'Samosa',
       'Biryani', 'Biryani', 'Pakora', 'Biryani', 'Samosa', 'Samosa',
       'Biryani', 'Biryani', 'Samosa', 'Biryani', 'Biryani', 'Samosa',
       'Pakora'], dtype=object)
```

In [11]:
```
from sklearn.metrics import accuracy_score

ac_s = accuracy_score(y_test, model)
ac_s
```

Out[11]:
```
0.5510204081632653
```

In [ ]:

In [ ]:

In [ ]: