NAME : SYED RIAZ ALI

EMAIL : syedriazali1997@gmail.com

WHATSAPP : 03002502513

Supervised Machine Learning (Classification : Decision Tree Classification)

In [36]:
```
# Importing libraries

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Loading data set

df = pd.read_csv("mldata_Day_21.csv")
df.head()
```

Out[36]:

	age	height	weight	gender	likeness
0	27	170.688	76.0	Male	Biryani
1	41	165.000	70.0	Male	Biryani
2	29	171.000	80.0	Male	Biryani
3	27	173.000	102.0	Male	Biryani
4	29	164.000	67.0	Male	Biryani

In [37]:
```
# Making Categorical variable into numerical variable by simply assigning the value for each category

df["gender"] = df["gender"].replace("Male", 1)
df["gender"] = df["gender"].replace("Female", 1)
df.head()
```

Out[37]:

	age	height	weight	gender	likeness
0	27	170.688	76.0	1	Biryani
1	41	165.000	70.0	1	Biryani
2	29	171.000	80.0	1	Biryani
3	27	173.000	102.0	1	Biryani
4	29	164.000	67.0	1	Biryani

In [18]:
```
# Selection of Input Variables

X = df[["weight","gender"]]
y = df["likeness"]
```

Training & Fitting the model

In [19]:
```
# Applying Machine Learning Algorithms
from sklearn.tree import DecisionTreeClassifier

# Creating and fit the model
model = DecisionTreeClassifier().fit(X, y)

# Prediction
model.predict([[80, 1]])
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

Out[19]:
```
array(['Biryani'], dtype=object)
```

Checking the accuracy of the model

In [20]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.2)

# Creating and fit the model
model = DecisionTreeClassifier().fit(X_train, y_train)

# Prediction
pred_values = model.predict(X_test)
pred_values
```

Out[20]:
```
array(['Pakora', 'Pakora', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Pakora', 'Samosa', 'Biryani', 'Biryani', 'Biryani', 'Pakora',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Pakora', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Pakora', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Biryani'], dtype=object)
```

In [22]:
```
# Checking Score

score = accuracy_score(y_test, pred_values)
score
```

Out[22]:
```
0.5510204081632653
```

How to train & Save your model

In [24]:
```
# How to train and save your model

import pandas as pd
from sklearn.tree import DecisionTreeClassifier
import joblib

model = DecisionTreeClassifier().fit(X, y)

joblib.dump(model, "foodie.joblib")
```

Out[24]:
```
['foodie.joblib']
```

Assignment : How to import & run a stored save model on our data ?

In [40]:
```
# After saving the model to a .joblib file we can load it by using the following command:

model_object = joblib.load("foodie.joblib")
```

In [41]:
```
# Now the model has been loaded you can make your prediction easily

weight_in_kg = int(input("Enter your weight in kgs : "))
gender_in_code = int(input("Enter your sex : If Male enter 1 and if Female enter 0 "))

x_pred_value = [[weight_in_kg, gender_in_code]]

# Making prediction by just calling the variable with .predict and just passing the values of input

model_object.predict(x_pred_value)
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

Out[41]:
```
array(['Biryani'], dtype=object)
```

Making a Tree Graph

In [26]:
```
# Tree Graph

from sklearn import tree

model = DecisionTreeClassifier().fit(X, y)

tree.export_graphviz(model,
                     out_file = "foodie.dot",
                     feature_names = ["age", "gender"],
                     class_names = sorted(y.unique()),
                     label = "all",
                     rounded = True,
                     filled = True)
```

Assignment : Train the model using all the variables in the data set

In [38]:
```
# Viewing the data set
df.head()
```

Out[38]:

	age	height	weight	gender	likeness
0	27	170.688	76.0	1	Biryani
1	41	165.000	70.0	1	Biryani
2	29	171.000	80.0	1	Biryani
3	27	173.000	102.0	1	Biryani
4	29	164.000	67.0	1	Biryani

In [39]:
```
# Selection of Input Variables

X = df[["age","height","weight","gender"]]
y = df["likeness"]
```

Training & Fitting the model

In [30]:
```
# Creating and fit the model
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier().fit(X, y)

# prediction

predicted_values = model.predict([[41, 165.0, 70,1]])
predicted_values
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

Out[30]:
```
array(['Biryani'], dtype=object)
```

Checking the accuracy of the model

In [31]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.2)

# Creating and fit the model
model = DecisionTreeClassifier().fit(X_train, y_train)

# Prediction
pred_values = model.predict(X_test)
pred_values
```

Out[31]:
```
array(['Samosa', 'Biryani', 'Samosa', 'Samosa', 'Samosa', 'Biryani',
       'Samosa', 'Biryani', 'Biryani', 'Biryani', 'Pakora', 'Biryani',
       'Samosa', 'Biryani', 'Samosa', 'Biryani', 'Biryani', 'Samosa',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Pakora', 'Biryani', 'Samosa', 'Biryani',
       'Pakora', 'Biryani', 'Pakora', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Pakora', 'Samosa', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Samosa',
       'Biryani'], dtype=object)
```

In [32]:
```
# Check score

score = accuracy_score(y_test, pred_values)
score
```

Out[32]:
```
0.5102040816326531
```

Making Tree Graph

In [34]:
```
# Tree Graph

from sklearn import tree

model = DecisionTreeClassifier().fit(X, y)

tree.export_graphviz(model,
                     out_file = "foodie_graph_2.dot",
                     feature_names = ["age", "gender","height", "weight"],
                     class_names = sorted(y.unique()),
                     label = "all",
                     rounded = True,
                     filled = True)
```

Saving model in to .joblib file

In [42]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
import joblib

model = DecisionTreeClassifier().fit(X, y)

joblib.dump(model, "foodie_2.joblib")
```

Out[42]:
```
['foodie_2.joblib']
```

Make prediction Game

Make your predictions on people favourite food Just ask that person 4 things and you will predict what food ("Biryani" , "Pakora", "Samosa") he/she likes

1. Weight of individual in kgs
2. Gender
   A. If Male enter 1
   B. If Female enter 0
3. Age in years
4. Height in centimeters

In [43]:
```
# After saving the model to a .joblib file we can load it by using the following command:

model_object = joblib.load("foodie_2.joblib")
```

In [44]:
```
# Now the model has been loaded you can make your prediction easily

weight_in_kg = int(input("Enter your weight in kgs : "))
gender_in_code = int(input("Enter your sex : If Male enter 1 and if Female enter 0 "))
age_in_years = int(input("Enter your age in years :"))
height_in_cm = int(input("Enter your height in centimeters :"))


x_pred_value = [[age_in_years, height_in_cm, weight_in_kg, gender_in_code]]

# Making prediction by just calling the variable with .predict and just passing the values of input

model_object.predict(x_pred_value)
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

Out[44]:
```
array(['Biryani'], dtype=object)
```

In [ ]:
```

```