

# Muhammad Arham Adeel

## Decision Tree Classifier

### Practice and assignment task in the video

```
In [1]: # import Libraries
```

```
import pandas as pd
```

```
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	165.000
2	29	80.0	Male	Biryani	171.000
3	27	102.0	Male	Biryani	173.000
4	29	67.0	Male	Biryani	164.000

```
In [3]: # here i can covert the gender column string to integer
df["gender"] = df["gender"].replace(["Male", "Female"], [1, 0])
# df["gender"] = df["gender"].replace("Female", 0)
```

```
In [4]: df.tail(10)
```

```
Out[4]:
```

	age	weight	gender	likeness	height
235	20	48.0	1	Biryani	155.0
236	30	50.0	0	Biryani	5.3
237	24	60.0	1	Biryani	114.0
238	22	50.0	1	Samosa	6.0
239	27	63.0	1	Biryani	165.0
240	31	60.0	1	Pakora	160.0
241	26	70.0	1	Biryani	172.0
242	40	80.0	1	Biryani	178.0
243	25	65.0	1	Biryani	5.7
244	33	56.0	0	Samosa	157.0

```
In [5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245 entries, 0 to 244
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         245 non-null   int64
1   weight      245 non-null   float64
2   gender      245 non-null   int64
3   likeness    245 non-null   object
4   height      245 non-null   float64
dtypes: float64(2), int64(2), object(1)
memory usage: 9.7+ KB
```

```
In [6]: df["height"].astype("int64")
```

```
Out[6]: 0      170
1      165
2      171
3      173
4      164
...
240    160
241    172
242    178
243      5
244    157
Name: height, Length: 245, dtype: int64
```

```
In [7]: df["gender"].value_counts()
```

```
Out[7]: 1      202
0       43
Name: gender, dtype: int64
```

## Selection and input variable

```
In [8]: x = df[["weight", "gender"]]
y = df["likeness"]
```

```
In [9]: x.head()
```

```
Out[9]:
```

	weight	gender
0	76.0	1
1	70.0	1
2	80.0	1
3	102.0	1
4	67.0	1

```
In [10]: y.head()
```

```
Out[10]: 0      Biryani
```

```
1 Biryani
2 Biryani
3 Biryani
4 Biryani
Name: likeness, dtype: object
```

## Machine Learning Algorithm

```
In [11]: from sklearn.tree import DecisionTreeClassifier

# Create and Fit our model

model = DecisionTreeClassifier().fit(x,y)
model
```

```
Out[11]: DecisionTreeClassifier()
```

```
In [12]: model.predict([[80,1]])
```

```
Out[12]: array(['Biryani'], dtype=object)
```

```
In [13]: model.predict([[23,0]])
```

```
Out[13]: array(['Biryani'], dtype=object)
```

```
In [14]: # how to measure the accuracy

#Split data in to train and test(80,20 rule)

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,random_state = 0) # 80% t
## Create a model and fitting a model
model = DecisionTreeClassifier()
model.fit(X_train,y_train)
predicted_values = model.predict(X_test)
```

```
In [15]: predicted_values
```

```
Out[15]: array(['Biryani', 'Biryani', 'Pakora', 'Biryani', 'Samosa', 'Biryani',
        'Pakora', 'Biryani', 'Biryani', 'Biryani', 'Samosa', 'Samosa',
        'Samosa', 'Pakora', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
        'Biryani', 'Pakora', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
        'Biryani', 'Biryani', 'Biryani', 'Samosa', 'Biryani', 'Samosa',
        'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
        'Biryani', 'Biryani', 'Samosa', 'Biryani', 'Biryani', 'Biryani',
        'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
        'Biryani'], dtype=object)
```

```
In [16]: #Checking Score

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

Out[16]: 0.6122448979591837

# How to train and save your model

```
In [17]: from sklearn.tree import DecisionTreeClassifier
import joblib

model1 = DecisionTreeClassifier().fit(x,y)

joblib.dump(model1,"train_and_save_your_model.joblib")
```

Out[17]: ['train\_and\_save\_your\_model.joblib']

In [18]:

Out[18]: DecisionTreeClassifier()

```
In [19]: #Graph

from sklearn import tree

model2 = DecisionTreeClassifier().fit(x,y)

# Graph Evaluation / Look into what happend?

tree.export_graphviz(model2, out_file = "foodie.dot", feature_names = ["age", "gender"],
                      class_names = sorted(y.unique()), label = "all", rounded = True, filled = True)
```

```
In [20]: x1 = df[["age", "weight", "gender", "height"]]
y = df["likeness"]
```

```
In [21]: from sklearn.tree import DecisionTreeClassifier

# Create and Fit our model

model = DecisionTreeClassifier().fit(x1,y)
model
```

Out[21]: DecisionTreeClassifier()

here i can take 4 columns and predict the the likeness base on age,weight, gender and height

```
In [22]: model.predict([[12,87,1,160]])
```

Out[22]: array(['Biryani'], dtype=object)

```
In [23]: model.predict([[60,48,0,145]])
```

array(['Pakora'], dtype=object)

Out[23]:

# how to import/run save model on our data?

In [28]:

```
# Load from file

joblib_LR_model = joblib.load("train_and_save_your_model.joblib")

joblib_LR_model
```

Out[28]:

DecisionTreeClassifier()

In [30]:

```
# Use the Reloaded Joblib Model to
# Calculate the accuracy score and predict target values

# Calculate the Score
score = joblib_LR_model.score(X_test, y_test)
# Print the Score

# Print the Score
print("Test score: {0:.2f} %".format(100 * score))
```

Test score: 79.59 %

In [ ]: