

Decision_tree_ML_03

March 15, 2022

```
[ ]: import pandas as pd
df = pd.read_csv('mldata.csv')
df.head()
```

```
[ ]:   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
```

```
[ ]: df['gender'] = df['gender'].replace("Male", 1)
df['gender'] = df['gender'].replace("Female", 0)
```

```
[ ]: #Slection of input and out puts
X = df[['weight', 'gender']]
y = df['likeness']
```

```
[ ]: X.head()
```

```
[ ]:   weight  gender
0    76.0      1
1    70.0      1
2    80.0      1
3   102.0      1
4    67.0      1
```

```
[ ]: #MAchine Learning algo
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier().fit(X,y)

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

```
C:\Users\Sartaj\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(
```

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

```
[ ]: model.predict([[80,0]])
```

```
C:\Users\Sartaj\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(
```

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

```
[ ]: # How to check accuracy  
# Split data  
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)
```

```
[ ]: model = DecisionTreeClassifier().fit(X_train,y_train)
```

```
[ ]: pridicted_value = model.predict(X_test)
```

```
[ ]: pridicted_value
```

```
[ ]: 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)
```

```
[ ]: #checking score  
score = accuracy_score(y_test, pridicted_value)  
score
```

```
[ ]: 0.6122448979591837
```

```
[ ]: #How to train and save model  
import joblib  
model = DecisionTreeClassifier().fit(X,y)  
  
joblib.dump(model,"foodiee.joblib")
```

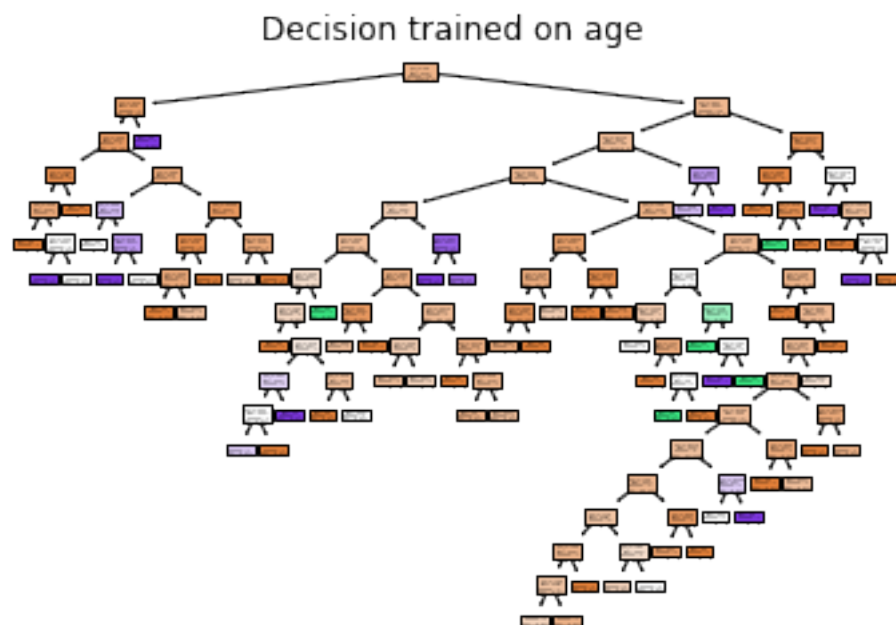
```
[ ]: ['foodiee.joblib']
```

```
[ ]: #How to import/run a stored model
loaded_model = joblib.load('foodiee.joblib')
result = loaded_model.score(X_test, y_test)
print(result)
```

0.7959183673469388

```
[ ]: #graph
from sklearn import tree
tree.export_graphviz(loaded_model,out_file='foodie.dot',
    ↳feature_names=['age','gender'],class_names=sorted(y.unique()), label='all',
    ↳rounded=True, filled=True)
```

```
[ ]: from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
p1 = plt.figure()
model = DecisionTreeClassifier().fit(X,y)
plot_tree(model, filled=True)
plt.title("Decision trained on age ")
plt.savefig("Decision tree.png",dpi=1500, facecolor='white',edgecolor='none')
plt.show()
plt.close()
```



```
[ ]: X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.3,
    ↳random_state=0)
model = DecisionTreeClassifier().fit(X_train,y_train)
```

```
pridicted_value = model.predict([[20,1]])
pridicted_value
```

C:\Users\Sartaj\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(

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

```
[ ]: #checking score
score = accuracy_score(y_test, pridicted_value)
score
```

```
[ ]: 0.6621621621621622
```

```
[ ]: X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.1,
↳random_state=0)
model = DecisionTreeClassifier().fit(X_train,y_train)
pridicted_value = model.predict(X_test)
pridicted_value
```

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

```
[ ]: #checking score
score = accuracy_score(y_test, pridicted_value)
score
```

```
[ ]: 0.6
```

```
[ ]: X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.5,
↳random_state=0)
model = DecisionTreeClassifier().fit(X_train,y_train)
pridicted_value = model.predict(X_test)
pridicted_value
```

```
[ ]: array(['Pakora', 'Biryani', 'Pakora', 'Biryani', 'Biryani', 'Biryani',
'Pakora', 'Biryani', 'Biryani', 'Biryani', 'Samosa', 'Biryani',
'Biryani', 'Pakora', 'Biryani', 'Biryani', 'Samosa', 'Biryani',
'Biryani', 'Pakora', 'Biryani', 'Biryani', 'Samosa', 'Biryani',
'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Samosa',
'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Samosa',
'Biryani', 'Biryani', 'Samosa', 'Samosa', 'Biryani', 'Biryani',
'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Samosa', 'Pakora',
```

```
'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Samosa', 'Biryani',  
'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',  
'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',  
'Biryani', 'Biryani', 'Biryani', 'Samosa', 'Samosa', 'Pakora',  
'Pakora', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',  
'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',  
'Biryani', 'Biryani', 'Pakora', 'Biryani', 'Biryani', 'Biryani',  
'Biryani', 'Samosa', 'Samosa', 'Samosa', 'Biryani', 'Biryani',  
'Samosa', 'Biryani', 'Pakora', 'Biryani', 'Pakora', 'Biryani',  
'Biryani', 'Biryani', 'Samosa', 'Biryani', 'Biryani', 'Biryani',  
'Pakora', 'Samosa', 'Biryani', 'Biryani', 'Biryani', 'Biryani',  
'Samosa', 'Pakora', 'Biryani', 'Biryani', 'Biryani', 'Biryani',  
'Biryani', 'Biryani', 'Biryani'], dtype=object)
```

```
[ ]: #checking score  
score = accuracy_score(y_test, pridicted_value)  
score
```

```
[ ]: 0.5365853658536586
```