dtc_fooddata

January 27, 2022

[ ]:

```
import pandas as pd
df = pd.read_csv("mldata_dtc.csv")
df.head(1)
```

[ ]:

```
   age   height  weight gender likeness
0   27  170.688    76.0   Male  Biryani
```

0.0.1 Convert gender (M/F) to 1 and 0

[ ]:

```
df['gender'] = df['gender'].replace("Male",1)
df['gender'] = df['gender'].replace("Female",0)
df.tail(2)
```

[ ]:

```
     age  height  weight  gender likeness
243   25     5.7    65.0       1  Biryani
244   33   157.0    56.0       0   Samosa
```

[ ]:

```
X=df[['weight','gender','age','height']]
#print("the value in X feature is ",X.head(3))
y=df['likeness']
#print("the value in y output  is ",y.head(3))
```

[ ]:

```
#machine learning algorithm
from sklearn.tree import DecisionTreeClassifier
# create and fit model
model = DecisionTreeClassifier().fit(X,y)
#Prediction
model.predict([[23,0,23,171]])
```

```
C:\Users\dell7450\AppData\Local\Programs\Python\Python310\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(['Pakora'], dtype=object)
```

0.0.2 How to measure accuracy (SPlit 80-20)

[ ]:

```
# accuracy by splitting
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# split syntax
X_train,X_test,y_train,y_test =train_test_split(X, y, test_size=0.2)
#Creating and model fitting
model = DecisionTreeClassifier().fit(X_train,y_train)
# checking predicted values with input test data
predicted_values = model.predict(X_test)
print("The predicted values from 20% of test input is",predicted_values,"\n")
```

```
The predicted values from 20% of test input is ['Biryani' 'Biryani' 'Biryani'
'Pakora' 'Biryani' 'Samosa' 'Biryani'
 'Samosa' 'Biryani' 'Samosa' 'Pakora' 'Biryani' 'Biryani' 'Biryani'
 'Samosa' 'Biryani' 'Biryani' 'Biryani' 'Biryani' 'Biryani' 'Biryani'
 'Biryani' 'Biryani' 'Samosa' 'Biryani' 'Biryani' 'Biryani' 'Biryani'
 'Pakora' 'Biryani' 'Biryani' 'Samosa' 'Pakora' 'Samosa' 'Biryani'
 'Samosa' 'Pakora' 'Biryani' 'Samosa' 'Biryani' 'Pakora' 'Pakora'
 'Biryani' 'Biryani' 'Biryani' 'Biryani' 'Biryani' 'Biryani' 'Biryani']
```

0.0.3 Score Checking

[ ]:

```
#Now compare y_test values with the values of y_test(predicted)
score = accuracy_score(y_test,predicted_values)
print("The accuracy score of model when compared with two test values is",score)
```

```
The accuracy score of model when compared with two test values is
0.4897959183673469
```

[ ]:

```
#graph
from sklearn import tree
model = DecisionTreeClassifier().fit(X,y)

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

0.0.4 How to train and save our Model

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

     model = DecisionTreeClassifier().fit(X,y)
     joblib.dump(model,"foodie.joblib")

     # How to run save stored model (Assignment)
     saved_model=joblib.load('foodie.joblib')

     Final_predictions=saved_model.predict(X_test)
     Final_predictions
```

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

```
[ ]:
```