In [ ]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from sklearn.tree import DecisionTreeClassifier, export_graphviz
import graphviz
import warnings
```

In [ ]:
```
warnings.filterwarnings("ignore")
data = pd.read_csv('features.csv')
data.head()
```

Out[ ]:

	label	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	Pitch	Roll	Yaw	AUs1	...	x_63	y_63	x_64	
0	Angry	-0.31715	-0.29662	36.38660	47.85770	0.99359	-9.59011	12.72161	-28.42508	0.29705	...	51.34946	49.90415	47.72598	47.
1	Angry	3.02009	-1.33738	42.36711	50.01836	0.95564	6.27818	1.67929	-0.37270	0.31780	...	57.55396	54.86165	50.38312	49.
2	Angry	0.98461	-1.29084	35.94039	47.50459	0.73074	-64.88136	-61.70474	70.54467	0.77023	...	48.56225	47.52279	46.07655	45.
3	Angry	5.36803	-0.93098	32.32775	44.01271	0.98712	-8.59568	-12.81338	0.20218	0.33113	...	45.59402	43.52846	39.25653	37.
4	Angry	2.61129	-0.59984	42.42387	48.76449	0.98044	6.68759	7.24873	-4.51140	0.36227	...	47.19149	46.04761	44.87098	44.

5 rows × 165 columns

In [ ]:
```
X = data.drop("label", axis=1)
y = data["label"]
X.head()
```

Out[ ]:

	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	Pitch	Roll	Yaw	AUs1	AUs2	...	x_63	y_63	x_64	
0	-0.31715	-0.29662	36.38660	47.85770	0.99359	-9.59011	12.72161	-28.42508	0.29705	0.11408	...	51.34946	49.90415	47.72598	4
1	3.02009	-1.33738	42.36711	50.01836	0.95564	6.27818	1.67929	-0.37270	0.31780	0.21831	...	57.55396	54.86165	50.38312	4
2	0.98461	-1.29084	35.94039	47.50459	0.73074	-64.88136	-61.70474	70.54467	0.77023	0.41684	...	48.56225	47.52279	46.07655	4
3	5.36803	-0.93098	32.32775	44.01271	0.98712	-8.59568	-12.81338	0.20218	0.33113	0.47253	...	45.59402	43.52846	39.25653	3
4	2.61129	-0.59984	42.42387	48.76449	0.98044	6.68759	7.24873	-4.51140	0.36227	0.28799	...	47.19149	46.04761	44.87098	4

5 rows × 164 columns

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

Out[ ]:
```
0    Angry
1    Angry
2    Angry
3    Angry
4    Angry
Name: label, dtype: object
```

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=y, random_state=42)
print(f"X_train shape: {X_train.shape}")
print(f"X_test shape: {X_test.shape}")
print(f"y_train shape: {y_train.shape}")
print(f"y_test shape: {y_test.shape}")
```

```
X_train shape: (122, 164)
X_test shape: (53, 164)
y_train shape: (122,)
y_test shape: (53,)
```

In [ ]:
```
classify = DecisionTreeClassifier(criterion='gini',max_leaf_nodes=8).fit(X_train,y_train)

data = export_graphviz(classify, out_file=None,
                           feature_names=X.columns,
                           class_names=classify.classes_,
                           filled=True, rounded=True,
                           special_characters=True)
graph=graphviz.Source(data)
graph.render(filename='graph', format='png', cleanup=True)
```

Out[ ]:
```
'graph.png'
```

In [ ]:
```
y_pred = classify.predict(X_test)
```

In [ ]:
```
print("Decision Tree Classifier")
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Decision Tree Classifier
              precision    recall  f1-score   support

       Angry       0.20      0.14      0.17         7
   Disgusted       0.00      0.00      0.00         7
        Fear       0.23      0.38      0.29         8
       Happy       0.80      0.50      0.62         8
     Neutral       0.30      0.38      0.33         8
         Sad       0.12      0.14      0.13         7
   Surprised       0.67      0.50      0.57         8

    accuracy                           0.30        53
   macro avg       0.33      0.29      0.30        53
weighted avg       0.34      0.30      0.31        53

Accuracy: 0.3018867924528302
```