Double-click (or enter) to edit

```
# LOading Data
```

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
music_data = pd.read_csv('/content/music.csv')
music_data
```

| | age | gender | genre | |
|----|--|---|---|---|
| 0 | 20 | 1 | HipHop | th |
| 1 | 23 | 1 | НірНор | +/ |
| 2 | 25 | 1 | НірНор | |
| 3 | 26 | 1 | Jazz | |
| 4 | 29 | 1 | Jazz | |
| 5 | 30 | 1 | Jazz | |
| 6 | 31 | 1 | Classical | |
| 7 | 33 | 1 | Classical | |
| 8 | 37 | 1 | Classical | |
| 9 | 20 | 0 | Dance | |
| 10 | 21 | 0 | Dance | |
| 11 | 25 | 0 | Dance | |
| 12 | 26 | 0 | Acoustic | |
| 13 | 27 | 0 | Acoustic | |
| 14 | 30 | 0 | Acoustic | |
| 15 | 31 | 0 | Classical | |
| 16 | 34 | 0 | Classical | |
| 17 | 35 | 0 | Classical | |
| | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 | 0 20 1 23 2 25 3 26 4 29 5 30 6 31 7 33 8 37 9 20 10 21 11 25 12 26 13 27 14 30 15 31 16 34 | 0 20 1 1 23 1 2 25 1 3 26 1 4 29 1 5 30 1 6 31 1 7 33 1 8 37 1 9 20 0 10 21 0 11 25 0 12 26 0 13 27 0 14 30 0 15 31 0 16 34 0 | 0 20 1 HipHop 1 23 1 HipHop 2 25 1 HipHop 3 26 1 Jazz 4 29 1 Jazz 5 30 1 Classical 7 33 1 Classical 8 37 1 Classical 9 20 0 Dance 10 21 0 Dance 11 25 0 Dance 12 26 0 Acoustic 13 27 0 Acoustic 14 30 0 Classical 15 31 0 Classical 16 34 0 Classical |

```
Next steps: Generate code with music_data View recommended plots New interactive sheet
```

```
# Prepaire the data of analysis
X = music_data.drop(columns=['genre'])
...
```

```
\overline{2}
          age gender
                          \blacksquare
       0
            20
                          ıl.
            23
       1
                     1
                          +0
       2
            25
                     1
       3
            26
                     1
            29
       4
                     1
       5
            30
            31
                     1
       6
            33
            37
       8
                     1
       9
            20
                     0
            21
                     0
      10
            25
      11
                     0
      12
            26
                     0
            27
      13
                     0
      14
            30
            3
      15
      16
            34
                     0
      17
            35
                     0
 Next steps: Generate co
                                      y = music_data['genre']
\overline{2}
             genre
       0
            HipHop
       1
            HipHop
       2
            HipHop
       3
       4
              Jazz
       5
              Jazz
          Classical
       6
       7
          Classical
       8
          Classical
       9
             Dance
      10
             Dance
      11
             Dance
      12
          Acoustic
      13
           Acoustic
      14
          Acoustic
      15 Classical
      16 Classical
      17 Classical
     dtype: object
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
# Create the model
model = DecisionTreeClassifier()
model.fit(X,y)
model.predict([[21,1],[22,0]])
\verb|predictions = model.predict([[21,1],[22,0]])|
predictions
# the accuracy of the model
model = DecisionTreeClassifier()
model.fit(X\_train,y\_train)
predictions = model.predict(X_test)
score = accuracy_score(y_test, predictions)
score
<del>→</del> 0.75
# persisting Models:
import joblib
joblib.dump(model, 'music-recommender.joblib')
→ ['music-recommender.joblib']
model = joblib.load('music-recommender.joblib')
predictions = model.predict([[21,1]])
predictions
🚁 /usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Deci
       warnings.warn(
     array(['HipHop'], dtype=object)
# Visualizing Decision Tree:
from sklearn import tree
tree.export_graphviz(model, out_file='music-recommender.dot',
                     feature_names=['age', 'gender'],
                     class_names=sorted(y.unique()),
                     label='all',
                     rounded=True,
                     filled=True)
import graphviz
with open("music-recommender.dot") as f:
    dot_graph = f.read()
graphviz.Source(dot_graph)
₹
                                                                  age <= 30.5
                                                                  gini = 0.796
                                                                  samples = 14
                                                              value = [2, 3, 3, 3, 3]
                                                                class = Classical
                                                             True
                                                                                 False
                                                   gender <= 0.5
                                                                                  gini = 0.0
                                                     gini = 0.744
                                                                                samples = 3
                                                    samples = 11
                                                                            value = [0, 3, 0, 0, 0]
                                                value = [2, 0, 3, 3, 3]
                                                                             class = Classical
                                                   class = Dance
                                                                  age <= 25.5
                                       age <= 26.0
                                       gini = 0.48
                                                                    gini = 0.5
                                       samples = 5
                                                                  samples = 6
                                  value = [2, 0, 3, 0, 0]
                                                              value = [0, 0, 0, 3, 3]
                                     class = Dance
                                                                 class = HipHop
```