In [15]:

```
# lets import a data set from kaggle (vgsales.csv)
    # import pandas as pd
    # df = pd.read_csv('vgsales.csv')
    # df.shape
```

In [16]:

```
# df.describe()
```

In [17]:

```
# df.values
```

In [13]:

```
# Jupyter Shortcuts
    # if you press h in the command mode(esc), we see the list of all the keyboard sho
```

In [ ]:

```
# A REAL PROBLEM
```

In [18]:

```
# First Step -- import data as csv
import pandas as pd
music_data = pd.read_csv('music.csv')
music_data
```

Out[18]:

	age	gender	genre
0	20	1	HipHop
1	23	1	HipHop
2	25	1	HipHop
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

In [ ]:

```
# Second Step --clean the data (we need to make an input set and output set)
# the output set, which is the genre column, contains the predictions
```

In [19]:

```
X = music_data.drop(columns = ['genre'])
X
```

Out[19]:

	age	gender
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
17	35	0

In [20]:

```
# next, we need to create output set
y = music_data['genre']
y
```

Out[20]:

```
0        HipHop
1        HipHop
2        HipHop
3          Jazz
4          Jazz
5          Jazz
6     Classical
7     Classical
8     Classical
9         Dance
10        Dance
11        Dance
12     Acoustic
13     Acoustic
14     Acoustic
15    Classical
16    Classical
17    Classical
Name: genre, dtype: object
```

In [24]:

```
# Fourth step --time to create a model (using an algorithm[decision tree])
import pandas as pd
from sklearn.tree import DecisionTreeClassifier

music_data = pd.read_csv('music.csv')
X = music_data.drop(columns = ['genre'])
y = music_data['genre']

model = DecisionTreeClassifier()
model.fit(X, y)
predictions = model.predict([ [21, 1], [22, 0] ])
predictions
```

Out[24]:

```
array(['HipHop', 'Dance'], dtype=object)
```

In [101…

```
# How do we measure the accuracy of the model?

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('music.csv')
X = music_data.drop(columns = ['genre'])
y = music_data['genre']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = DecisionTreeClassifier()
model.fit(X_train, y_train)
predictions = model.predict(X_test)

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

Out[101…

```
0.5
```

In [113…

```
# Persisting Models
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
import joblib

# music_data = pd.read_csv('music.csv')
# X = music_data.drop(columns = ['genre'])
# y = music_data['genre']

# model = DecisionTreeClassifier()
# model.fit(X, y)

model = joblib.load('music-recommender.joblib')
predictions = model.predict([[21, 1]])
predictions
```

Out[113…

```
array(['HipHop'], dtype=object)
```

In [112…

```
# Visualizing a Decision Tree
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree

music_data = pd.read_csv('music.csv')
X = music_data.drop(columns = ['genre'])
y = music_data['genre']

model = DecisionTreeClassifier()
model.fit(X, y)

tree.export_graphviz(model, out_file='music-recommender.dot',
                    feature_names=['age', 'gender'],
                    class_names=sorted(y.unique()),
                    label='all',
                    rounded=True,
                    filled=True)
```