Machine Learning to Predict Music Preference

Steps for Machine Learning

1. Import Data
2. Clean Data
3. Split the Data into Train/Test Sets
4. Create a Model
5. Train the Model
6. Make Predictions
7. Evaluate and Improve (Fine Tune or Use difference Algorithm)

```
from google.colab import files
uploaded = files.upload()
```

Choose Files music.csv

- **music.csv**(text/csv) - 247 bytes, last modified: 7/14/2023 - 100% done

```
Saving music.csv to music.csv
```

```
import pandas as pd
```

```
df = pd.read_csv("music.csv")
```

```
df.describe()
```

	age	gender
count	18.000000	18.000000
mean	27.944444	0.500000
std	5.127460	0.514496
min	20.000000	0.000000
25%	25.000000	0.000000
50%	28.000000	0.500000
75%	31.000000	1.000000
max	37.000000	1.000000

```
df
```

age gender genre

```
df.shape
```

```
(18, 3)
```

Split the Dataset into input set and target set

x is the class columns that are used to predict the y column

```
X = df.drop(columns=["genre"])
```

6 31 1 Classical

```
y = df['genre']
```

Training a model

```
from sklearn.tree import DecisionTreeClassifier
```

11 25 0 Dance

```
#Creating an instance of the DecisionTreeClassifier

model = DecisionTreeClassifier()


#Fit the data into the model
#.fit() takes two datasets, input set and target set

model.fit(X, y)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
#.predict() method takes a 2D array


#Here we are asking the model to make two predictions at the same time,
#one of 21 year old female, the other of 22 year old male



predictions = model.predict([ [21,1], [22, 0]])
predictions
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but DecisionTreeClas
  warnings.warn(
array(['HipHop', 'Dance'], dtype=object)
```

Measuring the accuracy of a model

We need to split the data into train set and test set

```
 from sklearn.model_selection import train_test_split


#train_test_split takes in 3 args

#The class variables

#The target variable

#Size of the test set


#retuns a tuple

#to unpack the tuple, we assign the train_test_split to four variables

#1 Input, Train

#2 Input, Test

#3 Target, Train
```

```
#4 Target, Test

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

This time we only fit the training datasets into the model

```
model.fit(X_train, y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

And, use the test datasets to test the model

```
predictions = model.predict(X_test)
```

Next we measure the accuracy of the predictions

```
from sklearn.metrics import accuracy_score
```

```
#accuracy score takes in 2 args,

#1 the actual target values

#2 the predicted taget values
```

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

```
1.0
```

Persisting a model that has been trained, so we dont have to train it everytime

```
from sklearn.externals import joblib

joblib.dump(model, 'music_recommender.joblib')
```

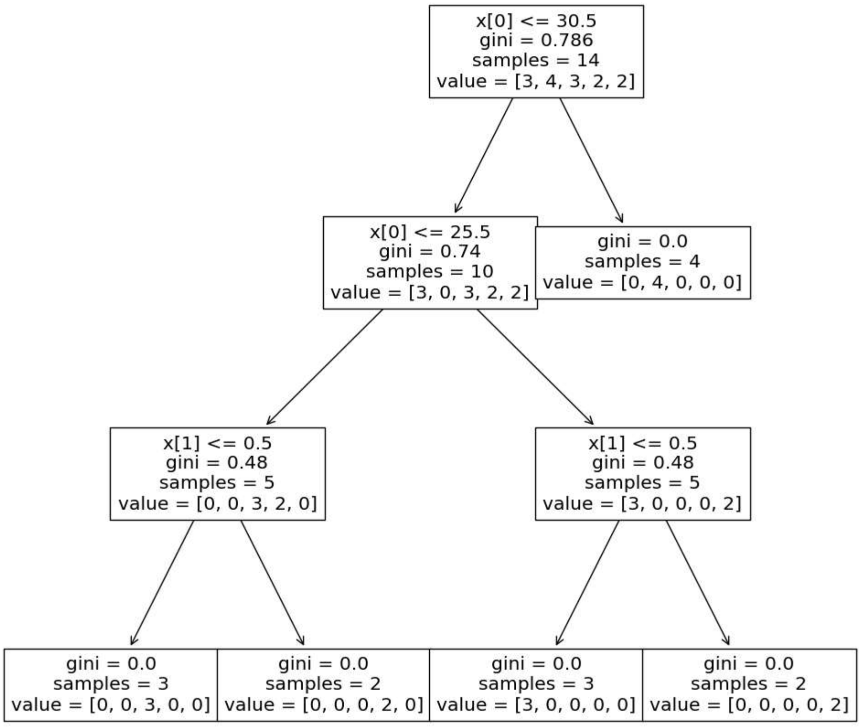
```
model = joblib.load('music_recommender.joblib')
```

Visualising the model

```
from sklearn import tree
import matplotlib.pyplot as plt
```

```
fig, ax = plt.subplots(figsize=(12, 12))
tree.plot_tree(model)

plt.show()
```



Colab paid products - Cancel contracts here

✓ 1s completed at 16:33