

# Real Machine Learning Project

In the reference video, the author presents a theoretical online music store that collects data (age, gender and preferred genre of music) from its customers and is looking to predict the preferred genre of music for new users. Below are the steps outlined in the video.

## Data Prepping

Here we import the data provided and split it into input and output.

```
In [1]: import pandas as pd
import warnings
warnings.filterwarnings('ignore')

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

```
In [2]: #Inspect X
X
```

```
Out[2]:
```

	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 [3]: #Inspect y
y.head()
```

```
Out[3]:
```

0	HipHop
1	HipHop
2	HipHop
3	Jazz
4	Jazz

Name: genre, dtype: object

# Learning and Predicting

Here we learn how to invoke and implement the decision tree module from the sklearn module.

```
In [4]: from sklearn.tree import DecisionTreeClassifier
```

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

```
Out[4]: array(['HipHop', 'Dance'], dtype=object)
```

## Determining the Accuracy of the model

In this section the video shows us how to calculate the accuracy of a decision tree model. It also discusses the train\_test\_split function and the impact of separating a % of the data set for testing on the accuracy outcomes.

```
In [5]: from sklearn.model_selection import train_test_split  
        from sklearn.metrics import accuracy_score
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)  
model.fit(X_train, y_train)  
predictions = model.predict(X_test)  
  
score = accuracy_score(y_test, predictions)  
score
```

```
Out[5]: 0.75
```

## Persisting Models

Here the video discusses how to use the model trained above without having to re-train the model each time.

```
In [6]: import joblib
```

```
joblib.dump(model, '../models/music-genre-predictor.joblib')
```

```
Out[6]: ['../models/music-genre-predictor.joblib']
```

```
In [7]: model = joblib.load('../models/music-genre-predictor.joblib')  
        predictions = model.predict([[21,1]])  
        predictions
```

```
Out[7]: array(['HipHop'], dtype=object)
```

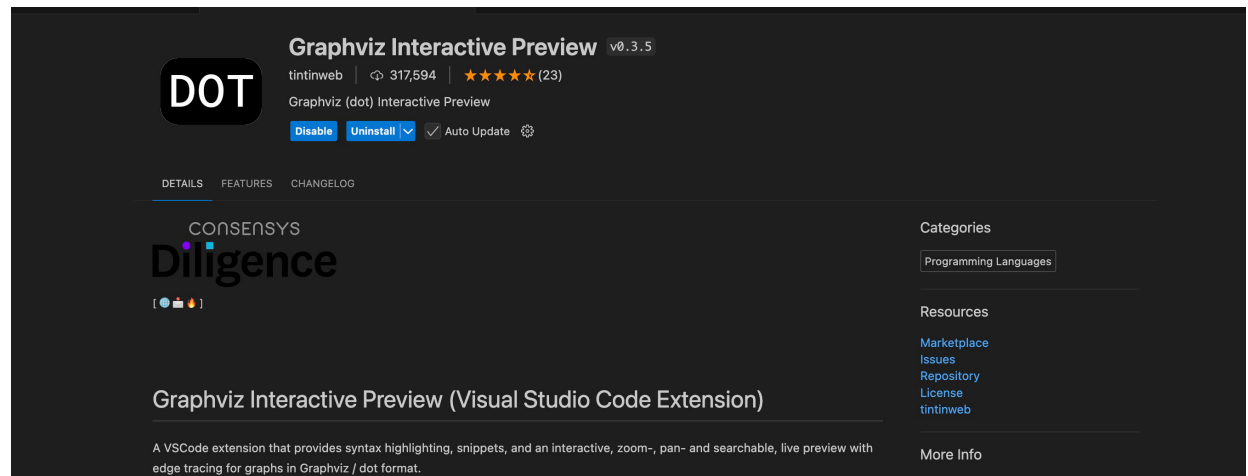
## Visualizing a Decision Tree

In this section we export a vizualization of the model of created above.

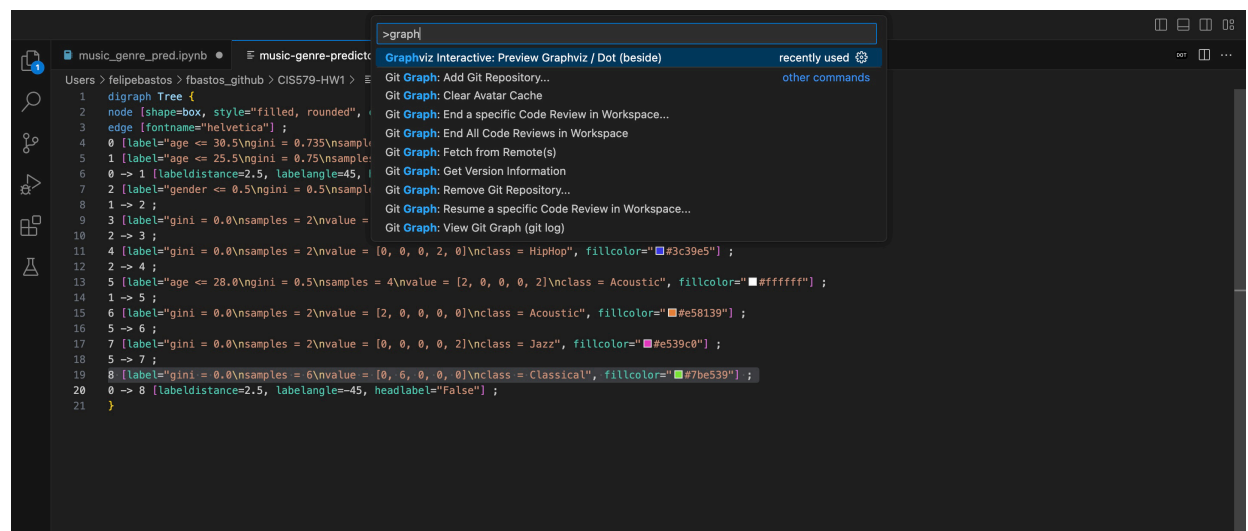
```
In [8]: from sklearn import tree  
        tree.export_graphviz(model,  
                             out_file='music-genre-predictor.dot',  
                             feature_names=['age', 'gender'],  
                             class_names=sorted(y.unique()),  
                             label='all',  
                             rounded=True,  
                             filled=True)
```

# VS code graph preview extension

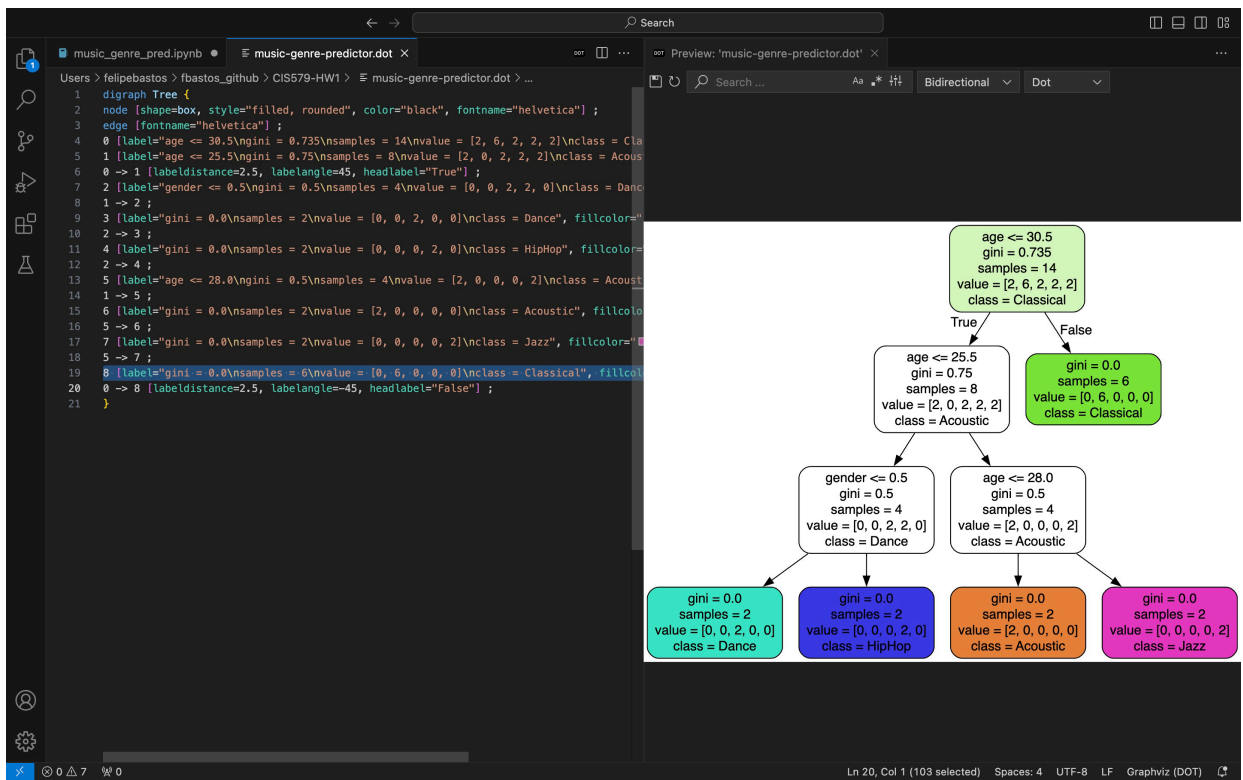
Since the release of the video the suggested extension no longer supports preview of .dot files. So a different extension had to be installed (see below)



To use the extension you need to open the .dot file in vs code and search ">graphviz Interactive: preview graphviz / Dot (beside)".



The above will result in the graph being displayed as follows:



Close up of resulting graph preview below:

