Machine Learning

www.hbpatel.in

Applications of Machine Learning

- Self-driving vehicles
- Robotics
- Language Processing
- Vision Processing
- Stock Market Prediction

Steps involved in Machine Learning

Steps of Machine Learning Project

- Import Data
- Clean Data
- Split Data (Training & Testing sets)
- Create a Model
- Train the Model
- Make Predictions
- Evaluate and Improve

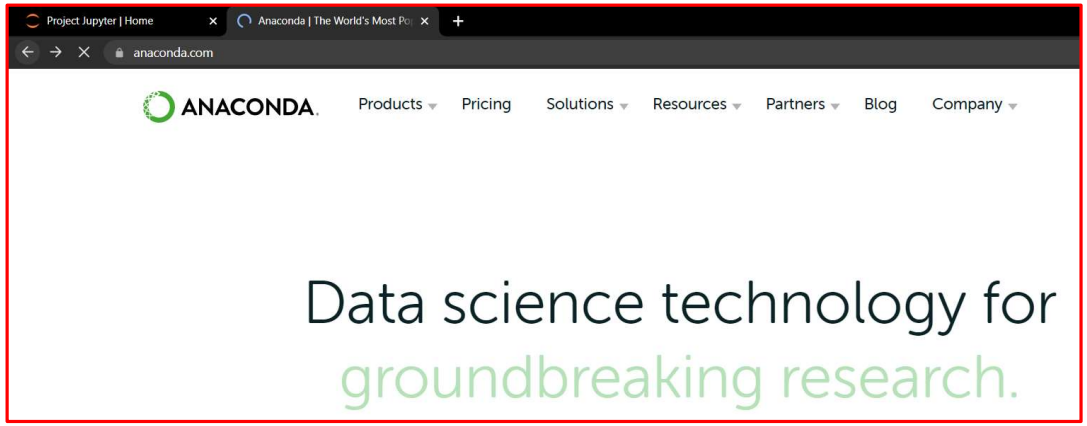
Machine Learning (Libraries and Tools)

www.hbpatel.in

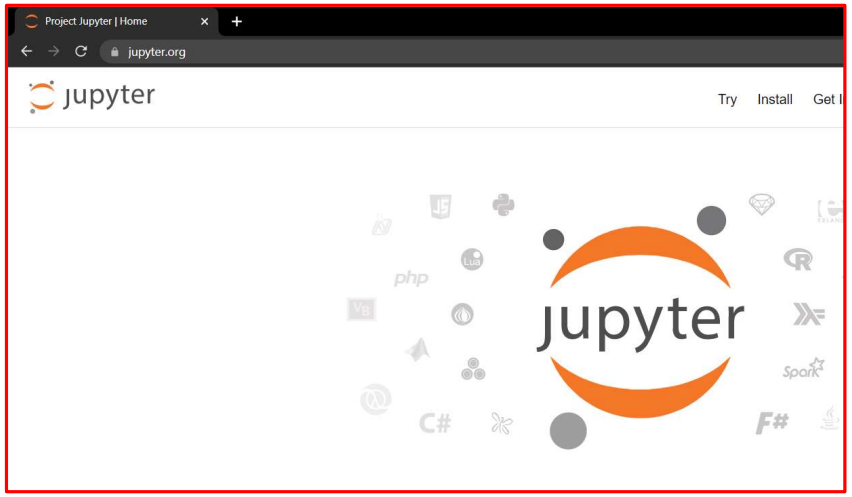
Libraries

- numpy
- Pandas
- Matplotlib
- Scikit-learn

To install Jupyter, we use the platform of Anaconda

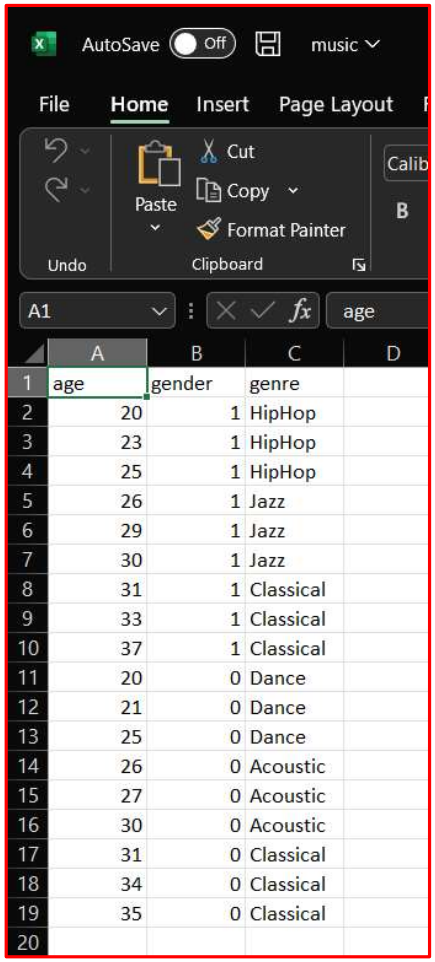


Jupyter Environment for Writing ML Project

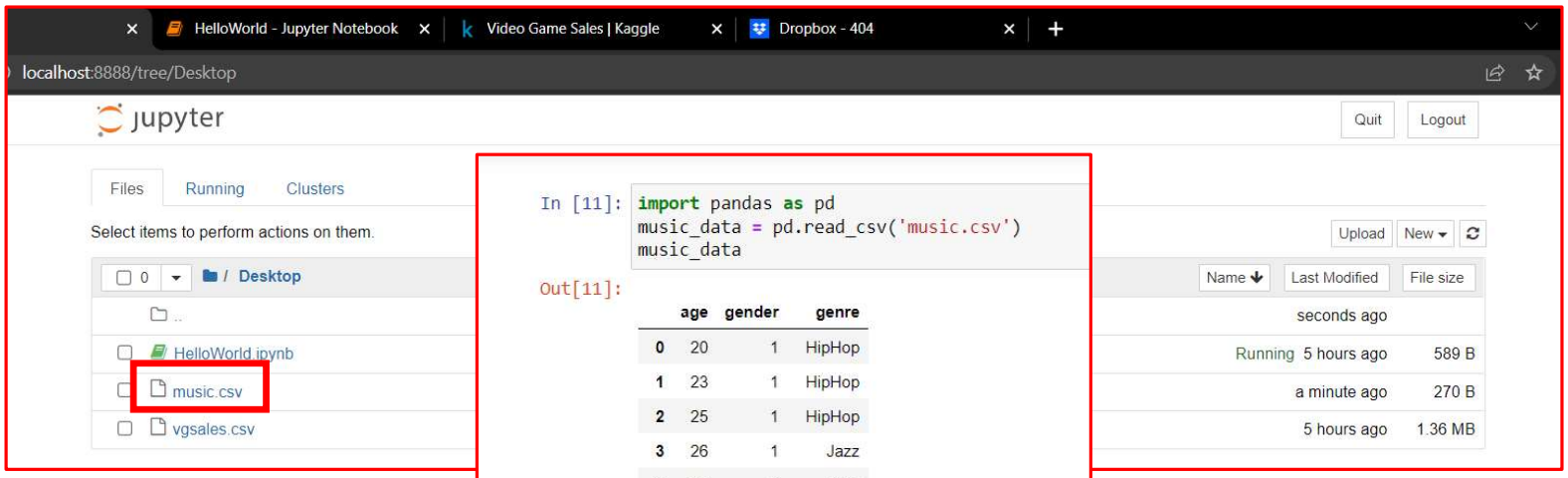


Step 1: Import Data

www.hbpatel.in



	A	B	C
1	age	gender	genre
2	20	1	HipHop
3	23	1	HipHop
4	25	1	HipHop
5	26	1	Jazz
6	29	1	Jazz
7	30	1	Jazz
8	31	1	Classical
9	33	1	Classical
10	37	1	Classical
11	20	0	Dance
12	21	0	Dance
13	25	0	Dance
14	26	0	Acoustic
15	27	0	Acoustic
16	30	0	Acoustic
17	31	0	Classical
18	34	0	Classical
19	35	0	Classical
20			




```
In [11]: import pandas as pd
         music_data = pd.read_csv('music.csv')
         music_data
```

Out[11]:

	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

Step 2: Clean Data

Remove duplicates and clean NULL values but we don't have such thing in our current music.csv file. However, we may split the database into two table, first containing the input (first two columns) and second containing the output (last column).

```
In [12]: import pandas as pd
         music_data = pd.read_csv('music.csv')
         x = music_data.drop(columns=['genre'])
         x
```

Out[12]:

	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

```
In [19]: import pandas as pd
         music_data = pd.read_csv('music.csv')
         x = music_data.drop(columns=['genre'])
         y = music_data['genre']
         y
```

```
Out[19]: 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
```

Step 3: Learning and Predicting Data

In [43]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier

#import datset
music_data = pd.read_csv('music.csv')
#create input set
X = music_data.drop(columns=['genre'])
#create output set
y = music_data['genre']

#create a model
model = DecisionTreeClassifier()
#train the model
model.fit(X, y)

music_data
```

Out[43]:

	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

In [25]:
```
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(col1, col2)

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

Out[25]: array(['HipHop', 'Dance'], dtype=object)

In [ ]:

Q: What does a male (1) of age (21) like? [21, 1]
Prediction: HipHop

Q: What does a female (0) of age (22) like? [22, 0]
Prediction: Dance

Step 4: Calculating Accuracy

Divide the dataset into two sets viz. (A) training dataset (70-80%), (B) testing dataset (20-30%)
Instead of passing one or two samples, we'll pass testing dataset and compare the resultant predictions with the actual values. Based on that, we can calculate the accuracy.
accuracy_score(y_test, predictions): y_test: Expected values, predictions: Actual values.

```
In [26]: 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[26]: 1.0
```

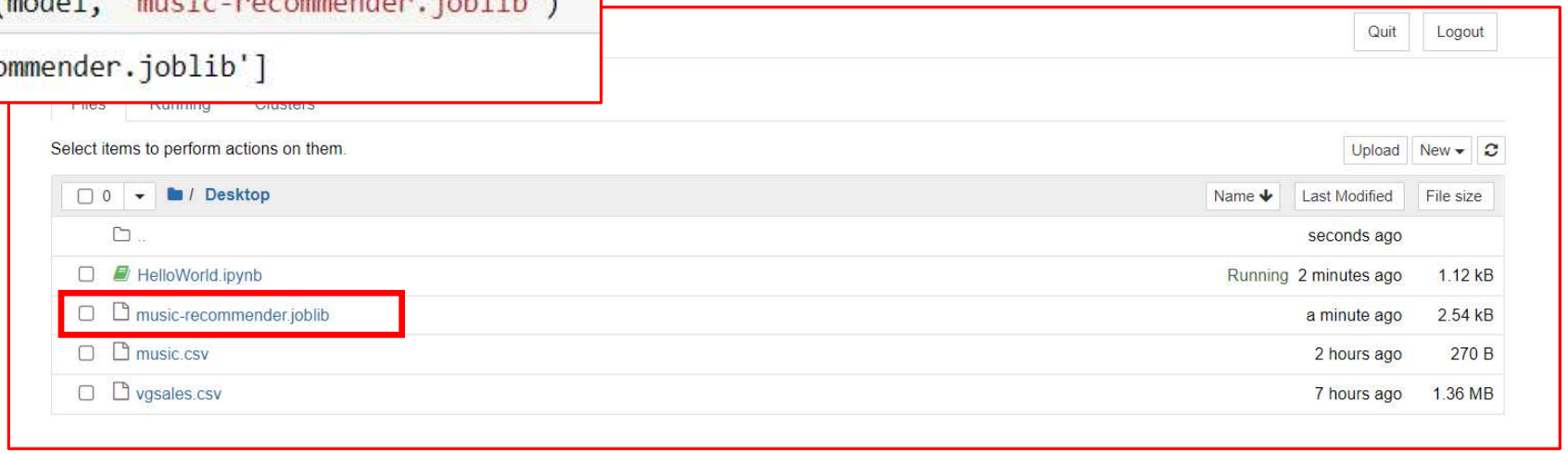
You may run this again and again with different size of training/testing data (E.g. test_size=0.5)

```
Out[29]: 0.75
```

Step 5: Persisting Models

In [39]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.externals 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)

joblib.dump(model, 'music-recommender.joblib')
```

Out[39]:
```
['music-recommender.joblib']
```



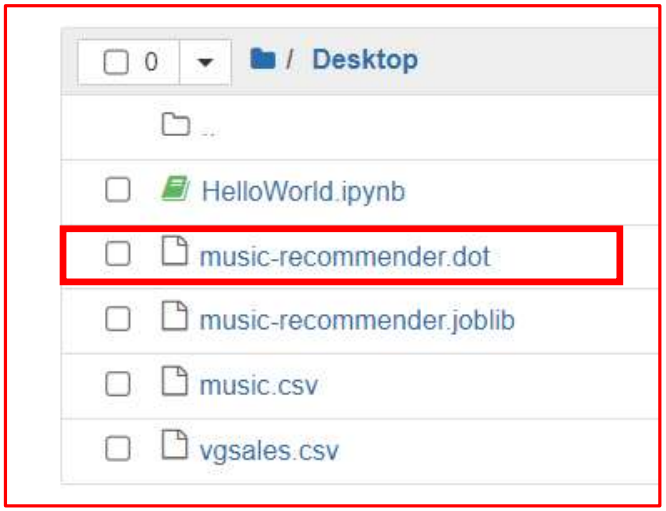
Step 5: Persisting Models

In [41]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.externals 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[41]:
```
array(['HipHop'], dtype=object)
```

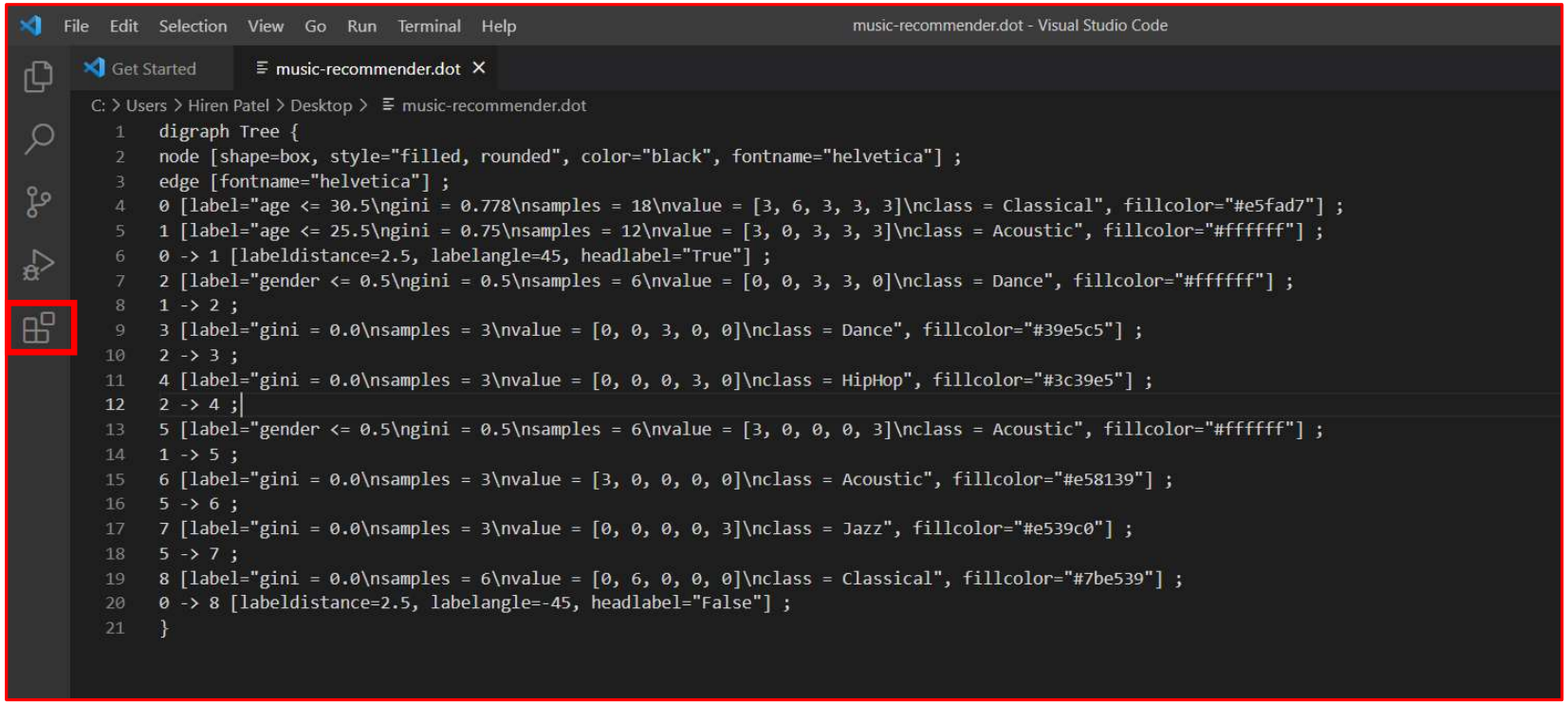
Step 6: Visualizing a Decision Tree

In [45]:
```
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)
```



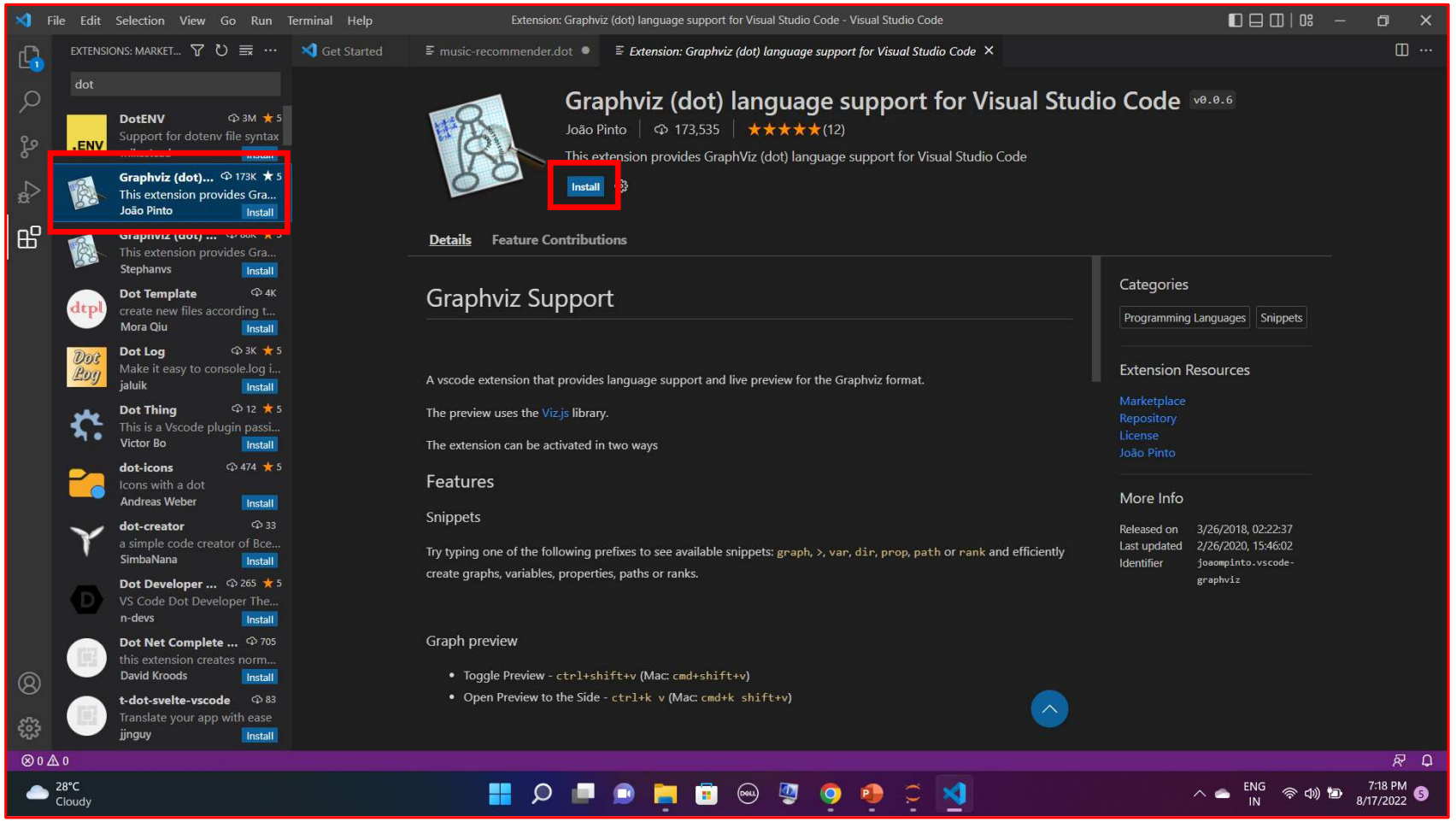
Step 6: Visualizing a Decision Tree

Open the music-recommender.dot [from C:\Users\Hiren Patel\Desktop] file in Visual Studio Code.



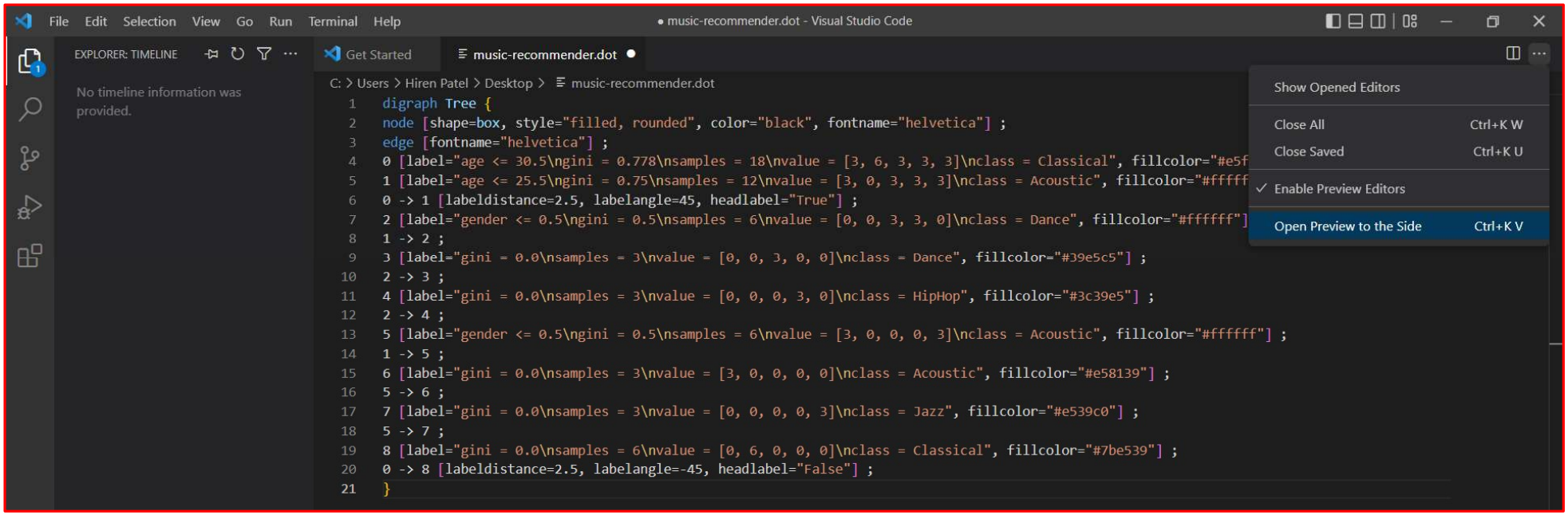
Step 6: Visualizing a Decision Tree

www.hbpatel.in



Step 6: Visualizing a Decision Tree

Reload Visual Studio Code.



Step 6: Visualizing a Decision Tree

www.hbpatel.in