

Age-based Music Playlist Prediction: Enhancing User Experience through Personalized Recommendations

Importing the data

Model used is Decision Tree Classifier

```
In [20]: import pandas as pd
        from sklearn.tree import DecisionTreeClassifier
        #!pip install --upgrade scikit-learn
```

To load the data

```
In [21]: music_data = pd.read_csv('C:\\Users\\HP\\Desktop\\VML with Mosh\\music.csv')
        music_data
```

Out[21]:

	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

creating only the input x (i.e) independent varibales

```
In [22]: x=music_data.drop(columns=['genre'])
        x
```

Out[22]:

	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

creating only the outputs y (i.e)predictions or Dependent variables

```
In [23]: y=music_data['genre']
        y
```

Out[23]:

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

Building a model using Decision tree classifier

```
In [24]: model = DecisionTreeClassifier()
        model.fit(x,y)
```

Out[24]:

DecisionTreeClassifier
DecisionTreeClassifier()

```
In [25]: # Making predictions with the Dependent and Independent variables
```

```
In [26]: # predictions = model.predict([[21,1],[22,0]])
        # predictions
```

```
In [27]: # Splitting the data and testing the accuracy
```

Importing the library for splitting Training and Testing

```
In [28]: from sklearn.model_selection import train_test_split
```

To calculate the accuracy

```
In [29]: from sklearn.metrics import accuracy_score
```

Splitting the data for Training and Testing

```
In [30]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2) # x_train,x_test input sets for training and testing
        # y_train,y_test output sets for training and testing
```

Training the model

```
In [31]: model.fit(x_train,y_train)
```

Out[31]:

DecisionTreeClassifier
DecisionTreeClassifier()

Performing predictions

```
In [32]: predictions = model.predict(x_test)
```

Testing the accuracy

```
In [33]: score = accuracy_score(y_test, predictions)
```

```
score
```

```
Out[33]: 1.0
```

Model persistance( for saving or Dumping and loading the model for prediction instead of training it every time)

for saving or Dumping a model

from sklearn.externals import joblib

```
In [ ]: import joblib
        joblib.dump(model,'music_recommender.joblib')
```

To load the model (i.e) the trained model

```
In [ ]: model = joblib.load('music_recommender.joblib')
```

Making predictions

```
In [39]: predictions = model.predict([[21,1]])
        predictions
```

C:\Users\HP\anaconda3\lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names

```
Out[39]: array(['HipHop'], dtype=object)
```

Graphical format

A method for transporting the Decision tree in a graphical format

```
In [ ]: 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)
```

Summary:

This project focuses on predicting users' preferred music playlists based on their age criteria.

Through the analysis of user data, including age and music preferences, the project aims to create a robust predictive model.

The model's objective is to accurately determine the type of playlist that aligns with a user's taste in music.

Furthermore, the project includes visualizing the model's results in a graphical format, making it easier to comprehend and interpret the playlist recommendations.

By providing personalized music recommendations, this project aims to enhance the overall music listening experience for users.