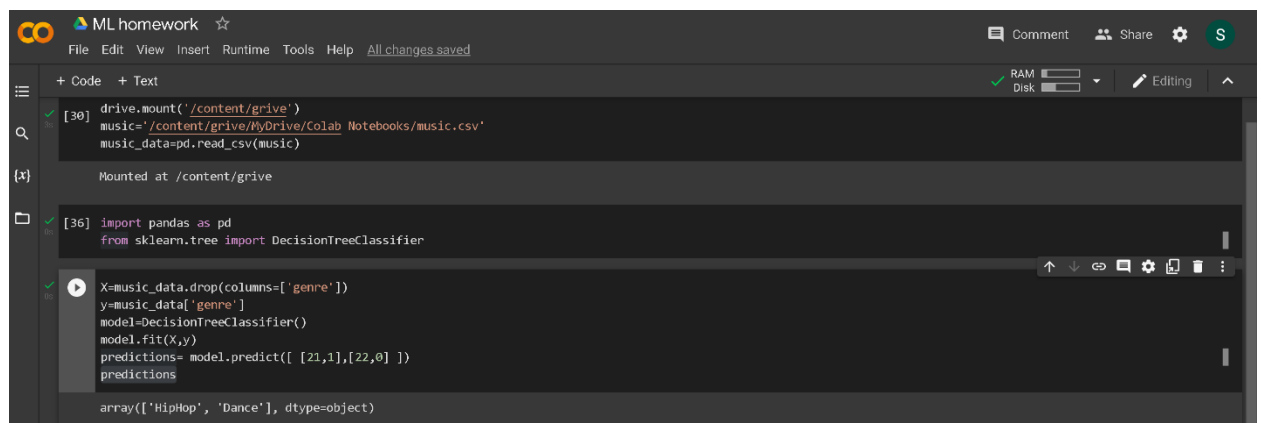


## ML Homework

In this machine learning assignment we will practice making a ML program that will recommend music albums users are willing to buy based on age and gender. This project is used to increase sales by taking existing profiles and applying that learned information to new users, then their data is used to suggest album genres. Methods applied are creating the algorithm model, training the model, and making predictions. Lastly, the models are evaluated and improved.

The problem the machine learning algorithm is trying to address is being able to accurately predict the genre of music for users given a specific age group and gender. This is useful information for sales, as stated above and is used to make informed decisions and suggestions for new users. Similarly to other music apps the more users and data that is received the more precise the music suggestions will be.

For this figure below we drop the data frame object we are use the parameter that specifies we drop the genre column in a new data set, represented by X. The output data set is y. This problem uses an algorithm called decision tree; we will use this to learn patterns. This model takes a two-dimensional array and forms a prediction. In which the program is able to identify that a 21-year-old male likes Hip-Hop, and a 21-year-old female likes Dance music. It is trying to predict what music type a person would listen to given age and gender.



```
ML homework
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
[30] drive.mount('/content/drive')
     music = '/content/drive/MyDrive/Colab Notebooks/music.csv'
     music_data = pd.read_csv(music)

Mounted at /content/drive

[36] import pandas as pd
     from sklearn.tree import DecisionTreeClassifier

X = music_data.drop(columns=['genre'])
y = music_data['genre']
model = DecisionTreeClassifier()
model.fit(X,y)
predictions = model.predict([ [21,1],[22,0] ])
predictions

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

In the two figures below we are measuring the accuracy of the models. We are trying to solve the problem of accuracy and making sure our ML algorithm is able to accurately and consistently make predictions. The data sets are split for training and testing. Normally 70-80% of our data is allocated to training and 20-30% is for testing. The testing data set will gather the predictions and then it will be compared to the actual values in the test set. The accuracy score between 0-1 will result. It was run twice, and the accuracy decreased from 1 to .5 due to the lack of data used to train the model.

```

[30] from google.colab import drive
drive.mount('/content/drive')
music='/content/drive/MyDrive/Colab Notebooks/music.csv'
music_data=pd.read_csv(music)

Mounted at /content/drive

[60] import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

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

1.0

```

*Table 1 sample size 0.2*

```

[30] from google.colab import drive
drive.mount('/content/drive')
music='/content/drive/MyDrive/Colab Notebooks/music.csv'
music_data=pd.read_csv(music)

Mounted at /content/drive

[60] import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

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=0.8)

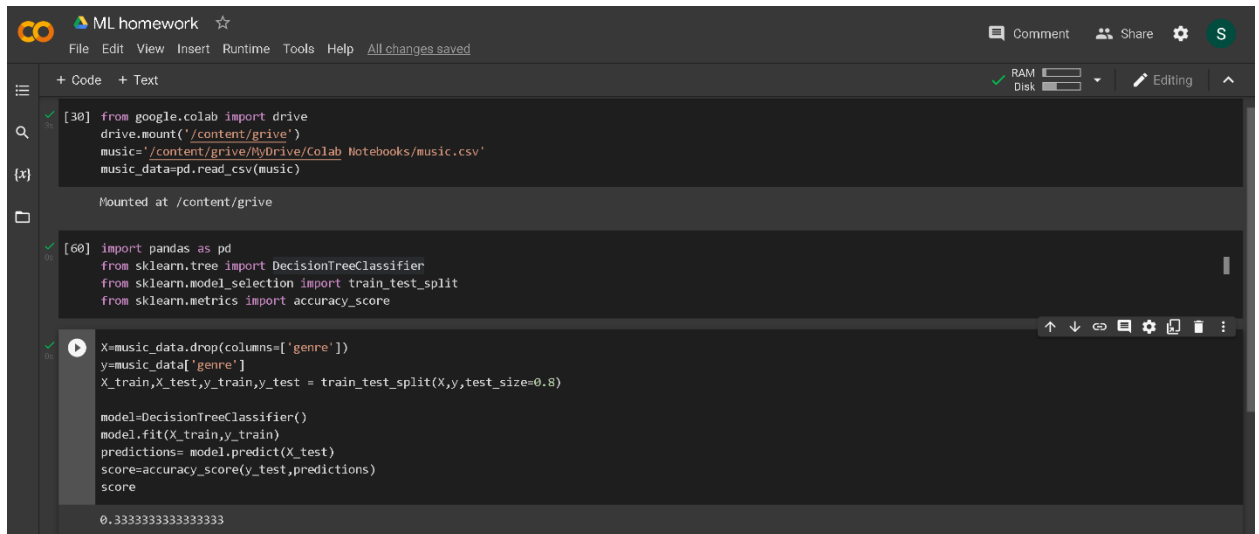
model=DecisionTreeClassifier()
model.fit(X_train,y_train)
predictions= model.predict(X_test)
score=accuracy_score(y_test,predictions)
score

0.5

```

*Table 2 sample size 0.8*

In the below table, the test size was changed from 0.2 to 0.8. Only 20% of the data was used for training the model and the other 80% is being used for testing. Once the cell is run multiple times the accuracy of the model is dropped, very little data is used to train the model. However, more data is typically needed to have a better result.



```
[30] from google.colab import drive
drive.mount('/content/drive')
music='/content/drive/MyDrive/Colab Notebooks/music.csv'
music_data=pd.read_csv(music)

Mounted at /content/drive

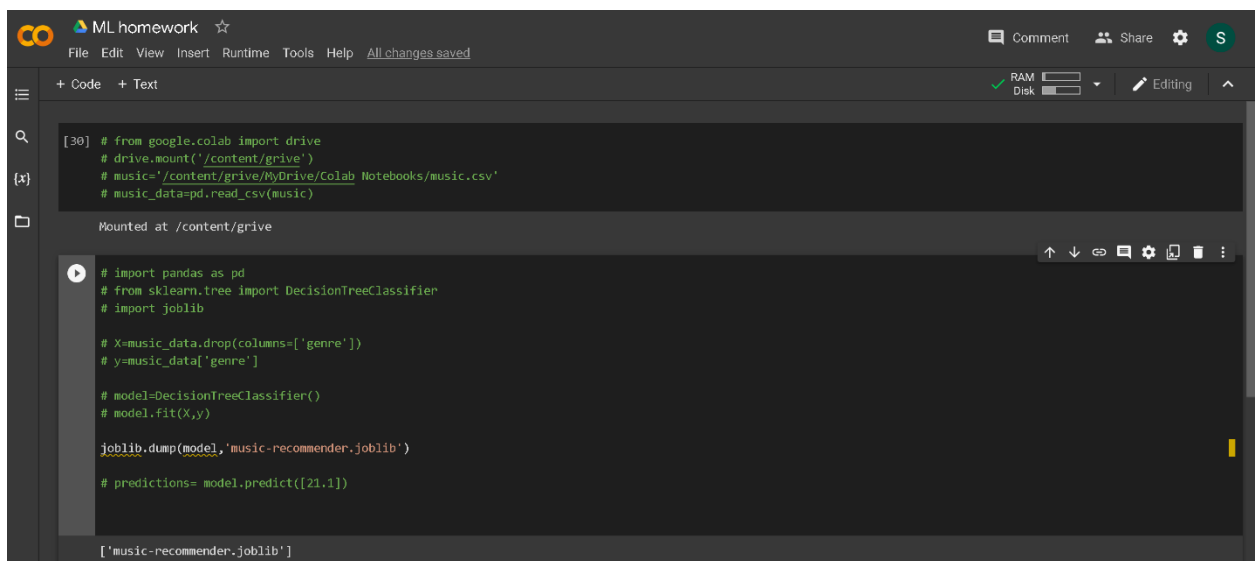
[60] import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

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=0.8)

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

0.3333333333333333
```

Joblib is a binary file used after a model has been trained. Joblib.dump is used when the trained model is stored. It is useful to use in circumstances where you don't want to train a model repeatedly.



```
[30] # from google.colab import drive
# drive.mount('/content/drive')
# music='/content/drive/MyDrive/Colab Notebooks/music.csv'
# music_data=pd.read_csv(music)

Mounted at /content/drive

# import pandas as pd
# from sklearn.tree import DecisionTreeClassifier
# import joblib

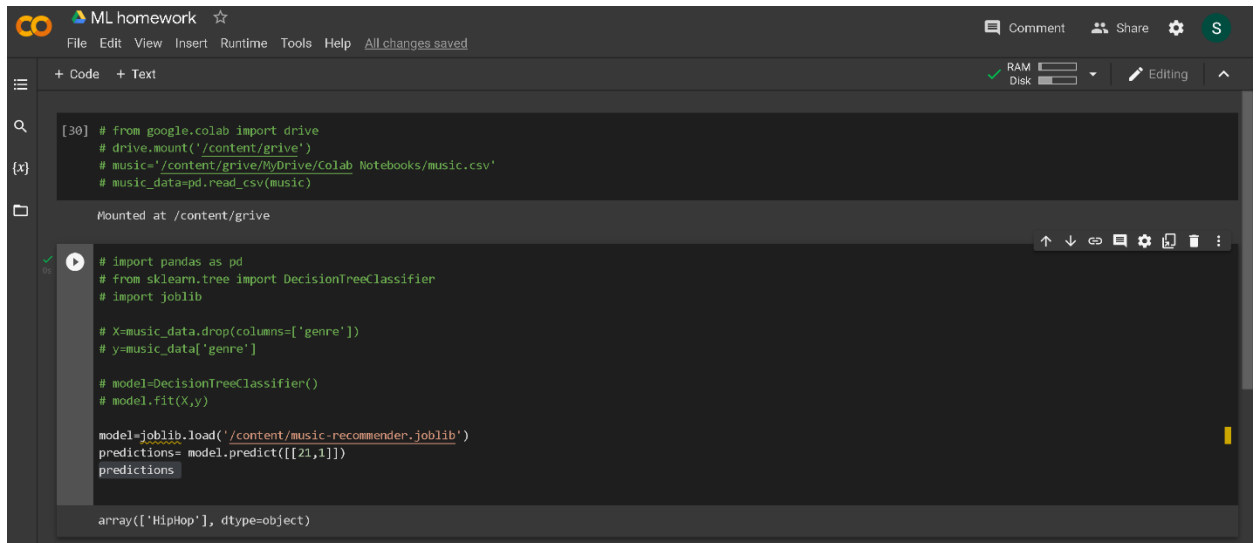
# X=music_data.drop(columns=['genre'])
# y=music_data['genre']

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

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

# predictions= model.predict([21.1])

['music-recommender.joblib']
```



```
[30] # from google.colab import drive
# drive.mount('/content/grive')
# music='/content/grive/MyDrive/Colab Notebooks/music.csv'
# music_data=pd.read_csv(music)

Mounted at /content/grive

# import pandas as pd
# from sklearn.tree import DecisionTreeClassifier
# import joblib

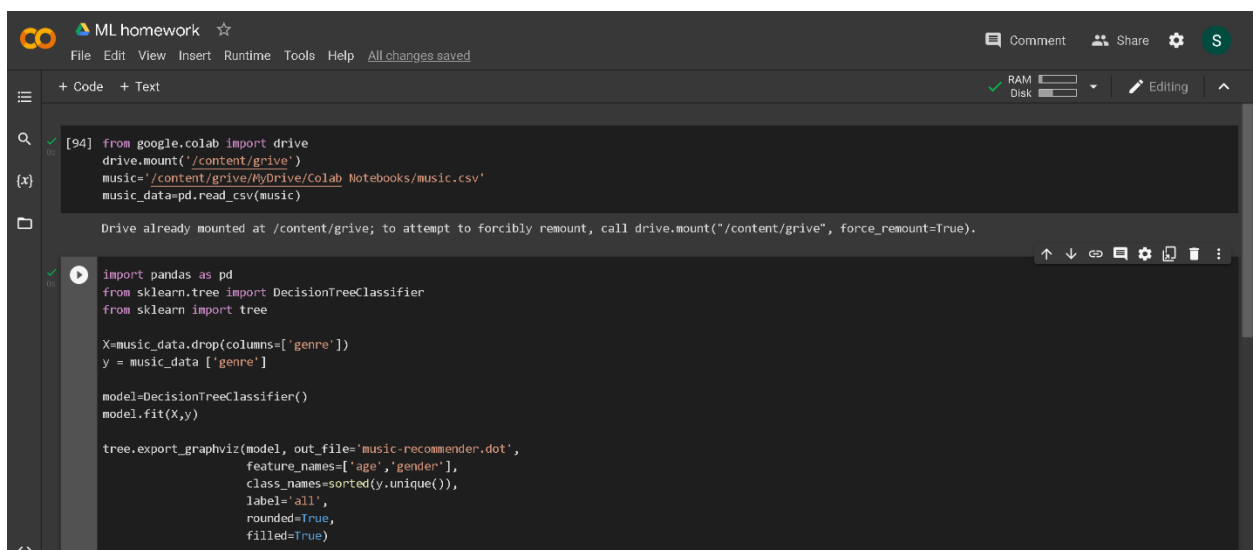
# X=music_data.drop(columns=['genre'])
# y=music_data['genre']

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

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

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

Table 3 Joblib was loaded instead of dumped. It loads the trained model, and predictions are able to be made.



```
[94] from google.colab import drive
drive.mount('/content/grive')
music='/content/grive/MyDrive/Colab Notebooks/music.csv'
music_data=pd.read_csv(music)

Drive already mounted at /content/grive; to attempt to forcibly remount, call drive.mount("/content/grive", force_remount=True).

import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree

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)
```

Table 4 dot for visualization

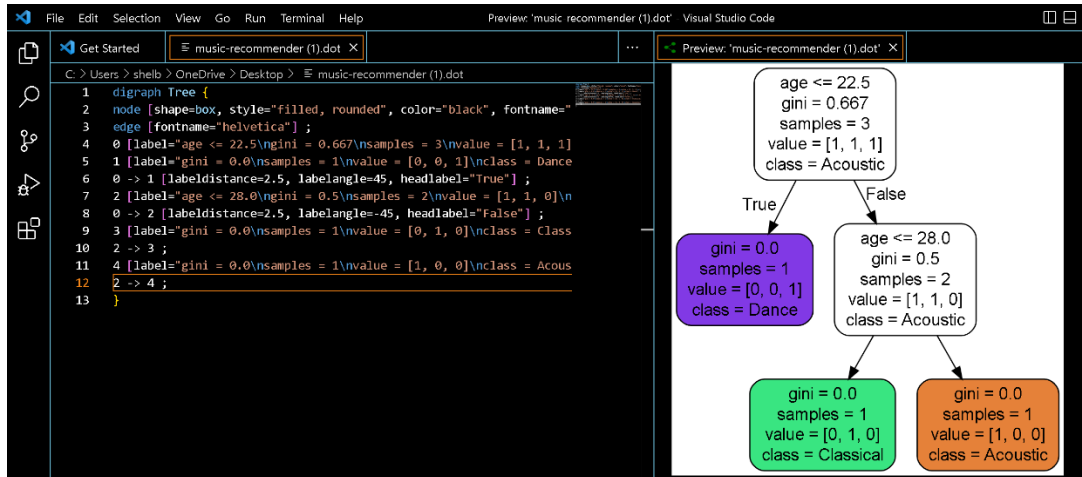


Table 5 dot file and visualization chart in VS

In this assignment we practiced making a ML program that recommends music albums to users that are willing to buy based on age and gender. The data visualization was also able to visually show the ML algorithm, through further data collecting and more testing, a more accurate visual representation would be achievable. The program was able to make predictions by taking existing profiles and applying that learned information to new users, then that data is used to suggest album genres. We were able to create the algorithm model, train the model, and make predictions. With this ML program, we were able to accurately predict the genre of music for users given a specific age group and gender.