SC1015
Mini Project

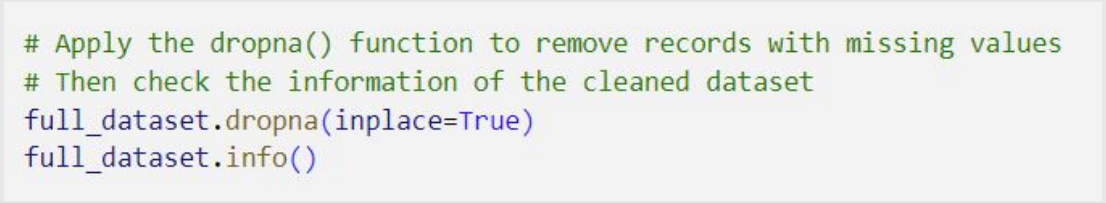
FDAB Group 3
U2323105G Zeng Zhuyu
U2321475L Zhang Yuxuan
U2323681A Tang Zhenwei

Table of contents

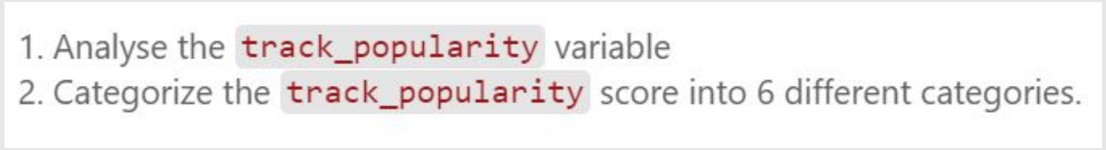
Motivation of our project

- Enhanced recommendation engine
- Sophisticated AI analysis
- Personalized user experience

Data cleaning and randomization

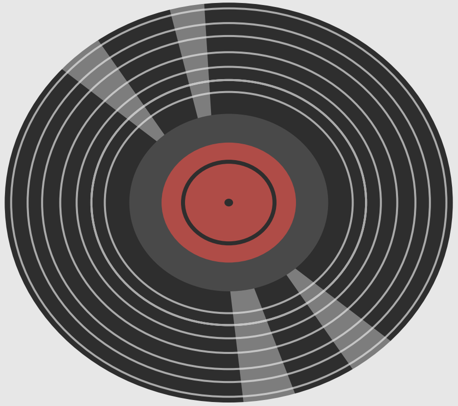


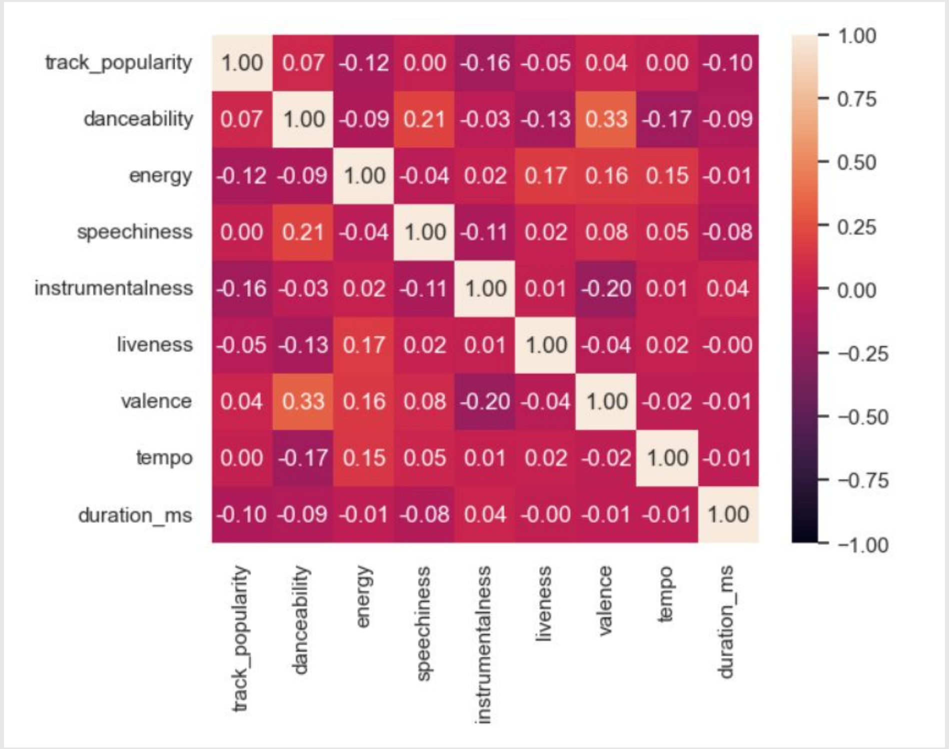
```
# Apply the dropna() function to remove records with missing values
# Then check the information of the cleaned dataset
full_dataset.dropna(inplace=True)
full_dataset.info()
```



1. Analyse the `track_popularity` variable
2. Categorize the `track_popularity` score into 6 different categories.

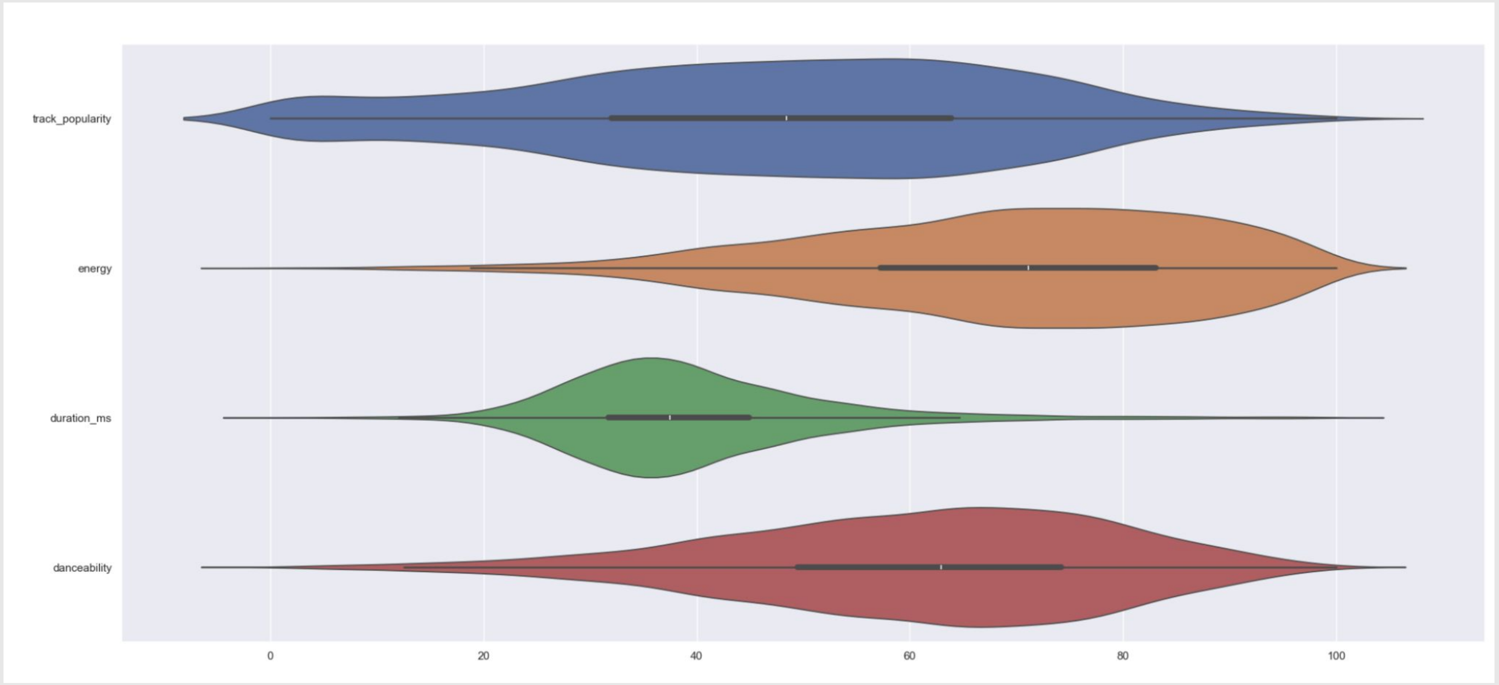
EDA & Visualization on numeric values



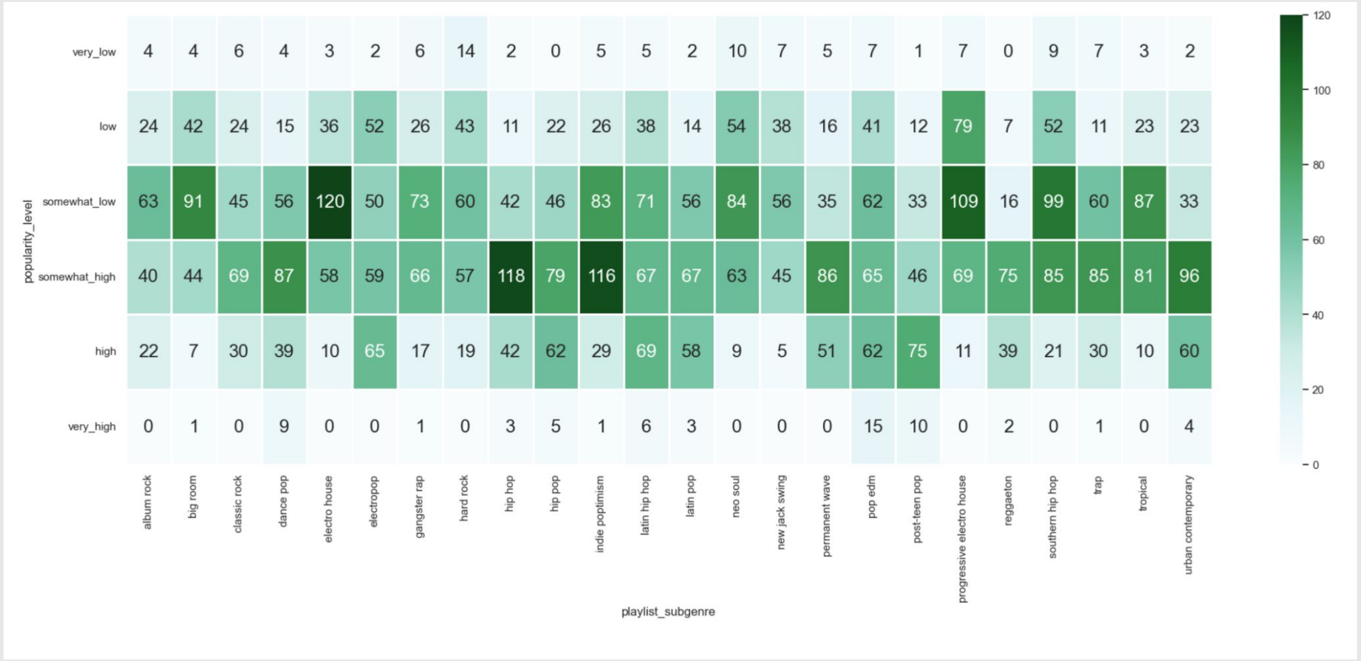


	track_popularity	danceability	energy	speechiness	instrumentalness	liveness	valence	tempo	duration_ms
track_popularity	1.00	0.07	-0.12	0.00	-0.16	-0.05	0.04	0.00	-0.10
danceability	0.07	1.00	-0.09	0.21	-0.03	-0.13	0.33	-0.17	-0.09
energy	-0.12	-0.09	1.00	-0.04	0.02	0.17	0.16	0.15	-0.01
speechiness	0.00	0.21	-0.04	1.00	-0.11	0.02	0.08	0.05	-0.08
instrumentalness	-0.16	-0.03	0.02	-0.11	1.00	0.01	-0.20	0.01	0.04
liveness	-0.05	-0.13	0.17	0.02	0.01	1.00	-0.04	0.02	-0.00
valence	0.04	0.33	0.16	0.08	-0.20	-0.04	1.00	-0.02	-0.01
tempo	0.00	-0.17	0.15	0.05	0.01	0.02	-0.02	1.00	-0.01
duration_ms	-0.10	-0.09	-0.01	-0.08	0.04	-0.00	-0.01	-0.01	1.00

EDA & Visualization on numeric values

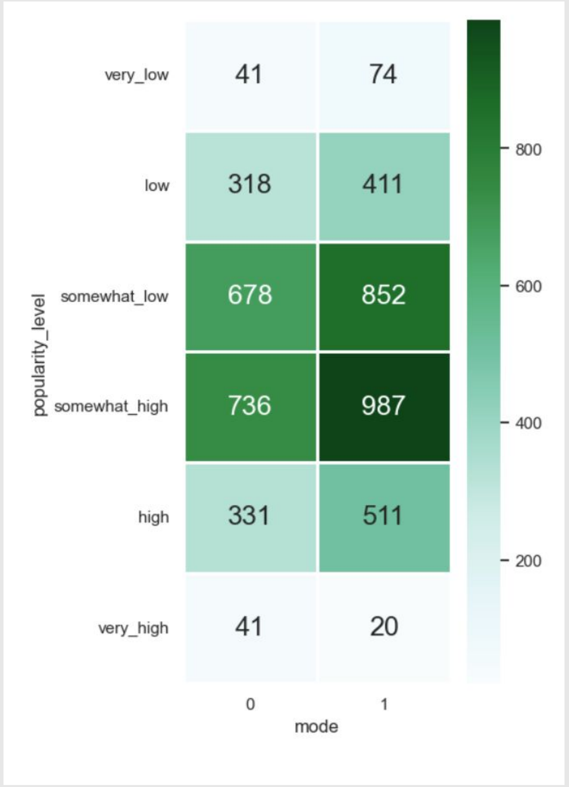


EDA & Visualization on categorical values



popularity_level	album rock	big room	classic rock	dance pop	electro house	electropop	gangster rap	hard rock	hip hop	hip pop	indie poptimism	latin hip hop	latin pop	neo soul	new jack swing	permanent wave	pop edm	post-teen pop	progressive electro house	reggaeton	southern hip hop	trap	tropical	urban contemporary
very_low	4	4	6	4	3	2	6	14	2	0	5	5	2	10	7	5	7	1	7	0	9	7	3	2
low	24	42	24	15	36	52	26	43	11	22	26	38	14	54	38	16	41	12	79	7	52	11	23	23
somewhat_low	63	91	45	56	120	50	73	60	42	46	83	71	56	84	56	35	62	33	109	16	99	60	87	33
somewhat_high	40	44	69	87	58	59	66	57	118	79	116	67	67	63	45	86	65	46	69	75	85	85	81	96
high	22	7	30	39	10	65	17	19	42	62	29	69	58	9	5	51	62	75	11	39	21	30	10	60
very_high	0	1	0	9	0	0	1	0	3	5	1	6	3	0	0	0	15	10	0	2	0	1	0	4

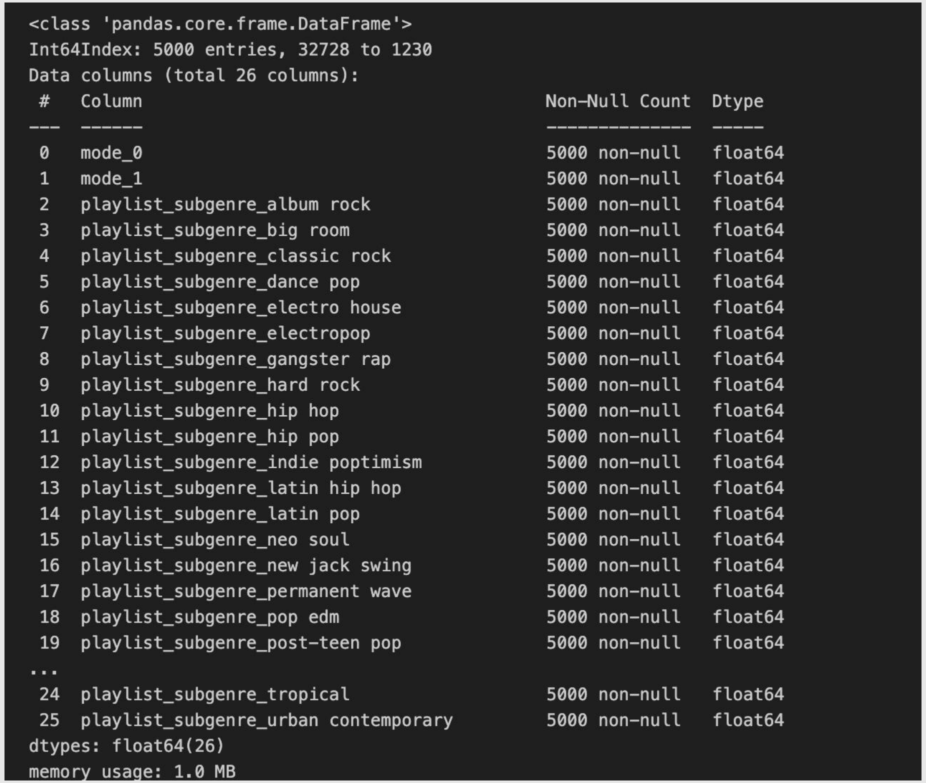
EDA & Visualization on categorical values



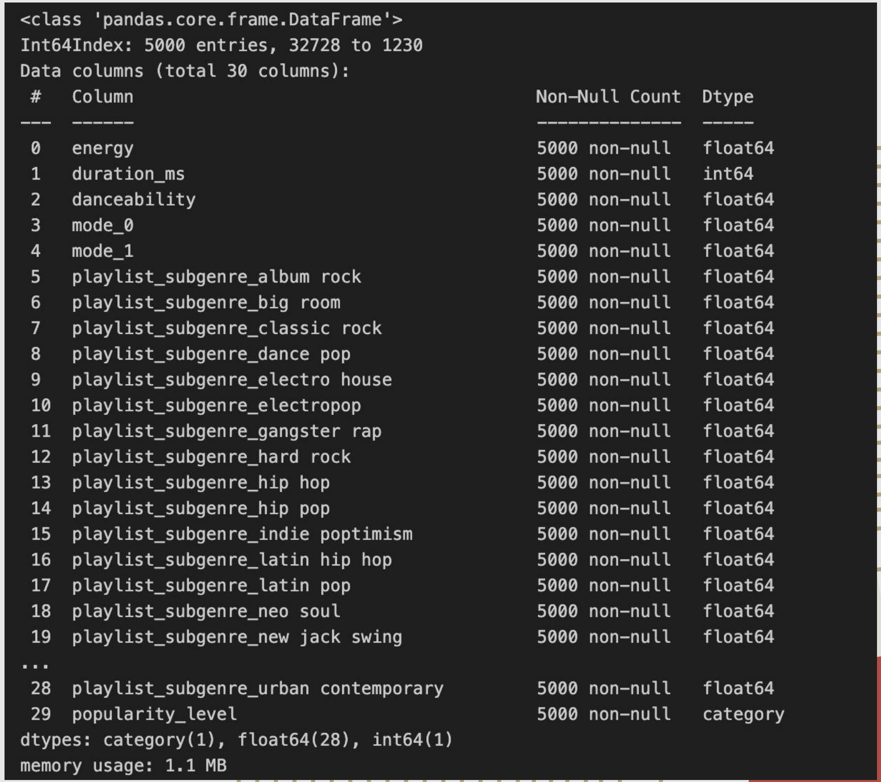
Model 1

For prediction of popularity_level

Pre-process data

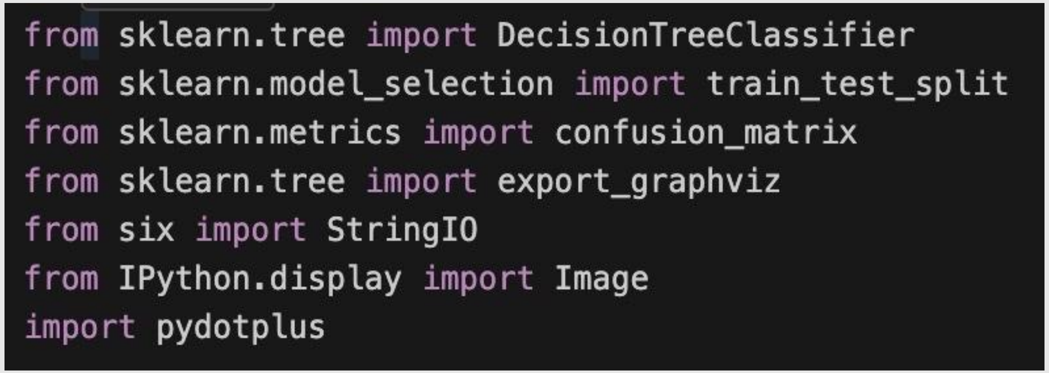


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5000 entries, 32728 to 1230
Data columns (total 26 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   mode_0                                5000 non-null   float64
 1   mode_1                                5000 non-null   float64
 2   playlist_subgenre_album rock          5000 non-null   float64
 3   playlist_subgenre_big room            5000 non-null   float64
 4   playlist_subgenre_classic rock        5000 non-null   float64
 5   playlist_subgenre_dance pop           5000 non-null   float64
 6   playlist_subgenre_electro house       5000 non-null   float64
 7   playlist_subgenre_electropop          5000 non-null   float64
 8   playlist_subgenre_gangster rap        5000 non-null   float64
 9   playlist_subgenre_hard rock           5000 non-null   float64
 10  playlist_subgenre_hip hop             5000 non-null   float64
 11  playlist_subgenre_hip pop             5000 non-null   float64
 12  playlist_subgenre_indie poptimism     5000 non-null   float64
 13  playlist_subgenre_latin hip hop       5000 non-null   float64
 14  playlist_subgenre_latin pop           5000 non-null   float64
 15  playlist_subgenre_neo soul            5000 non-null   float64
 16  playlist_subgenre_new jack swing      5000 non-null   float64
 17  playlist_subgenre_permanent wave      5000 non-null   float64
 18  playlist_subgenre_pop edm             5000 non-null   float64
 19  playlist_subgenre_post-teen pop       5000 non-null   float64
...
 24  playlist_subgenre_tropical            5000 non-null   float64
 25  playlist_subgenre_urban contemporary  5000 non-null   float64
dtypes: float64(26)
memory usage: 1.0 MB
```

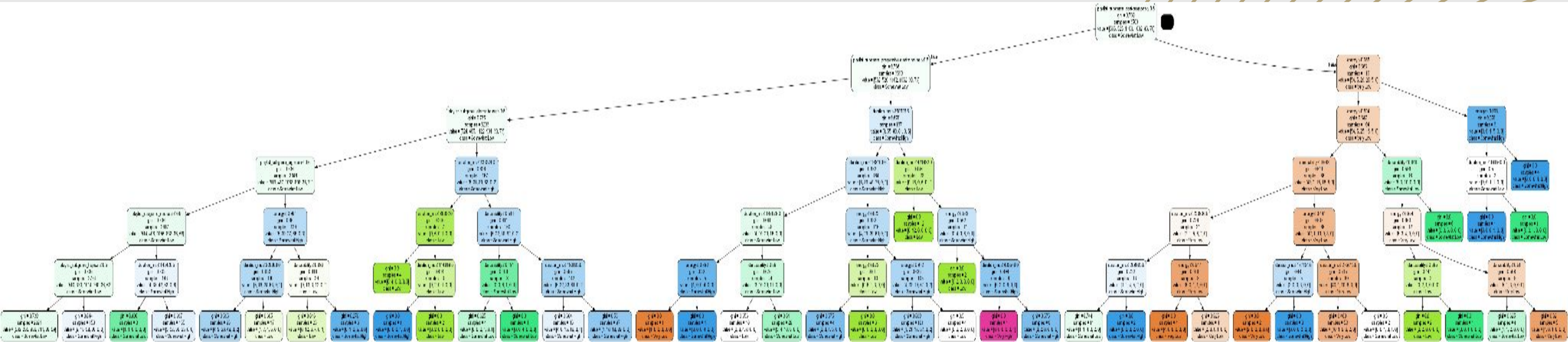


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5000 entries, 32728 to 1230
Data columns (total 30 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   energy                                5000 non-null   float64
 1   duration_ms                           5000 non-null   int64  
 2   danceability                          5000 non-null   float64
 3   mode_0                                5000 non-null   float64
 4   mode_1                                5000 non-null   float64
 5   playlist_subgenre_album rock          5000 non-null   float64
 6   playlist_subgenre_big room            5000 non-null   float64
 7   playlist_subgenre_classic rock        5000 non-null   float64
 8   playlist_subgenre_dance pop           5000 non-null   float64
 9   playlist_subgenre_electro house       5000 non-null   float64
 10  playlist_subgenre_electropop          5000 non-null   float64
 11  playlist_subgenre_gangster rap        5000 non-null   float64
 12  playlist_subgenre_hard rock           5000 non-null   float64
 13  playlist_subgenre_hip hop             5000 non-null   float64
 14  playlist_subgenre_hip pop             5000 non-null   float64
 15  playlist_subgenre_indie poptimism     5000 non-null   float64
 16  playlist_subgenre_latin hip hop       5000 non-null   float64
 17  playlist_subgenre_latin pop           5000 non-null   float64
 18  playlist_subgenre_neo soul            5000 non-null   float64
 19  playlist_subgenre_new jack swing      5000 non-null   float64
...
 28  playlist_subgenre_urban contemporary  5000 non-null   float64
 29  popularity_level                      5000 non-null   category
dtypes: category(1), float64(28), int64(1)
memory usage: 1.1 MB
```

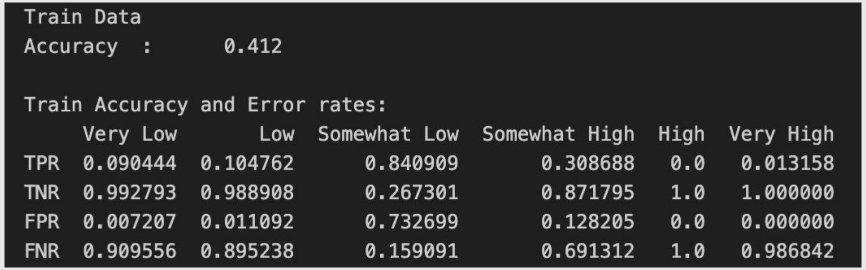
Create and fit dectree_1



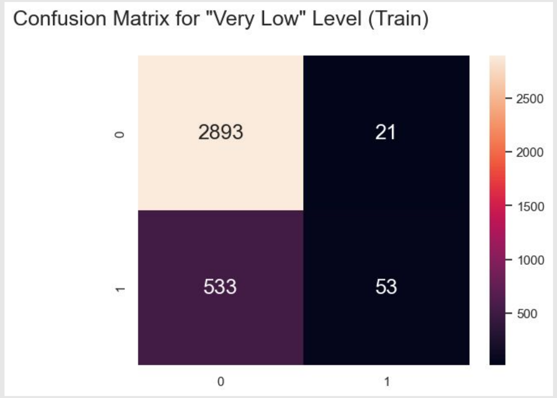
```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.tree import export_graphviz
from six import StringIO
from IPython.display import Image
import pydotplus
```

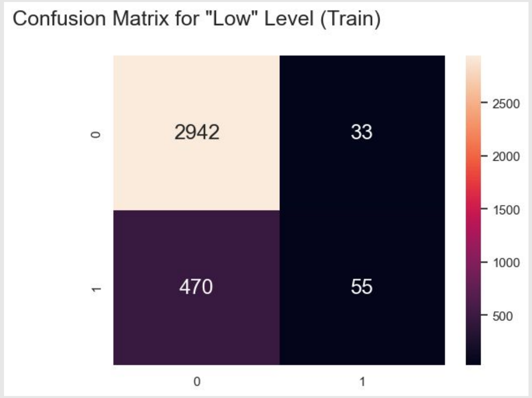


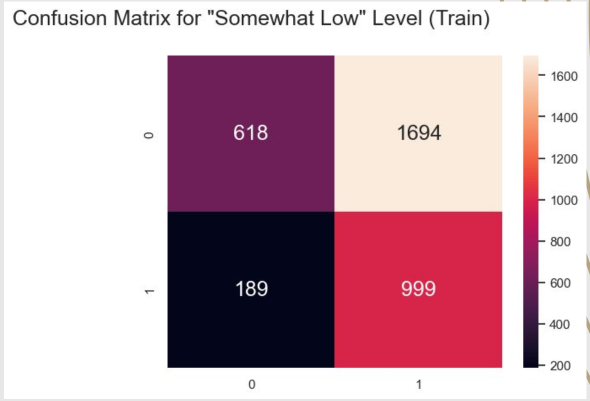
Check the accuracy of dectree_1



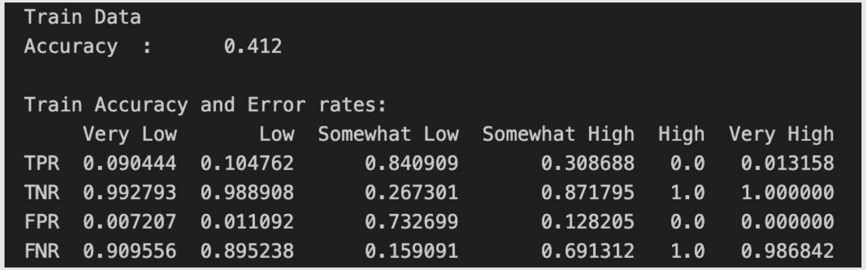
```
Train Data
Accuracy  :      0.412

Train Accuracy and Error rates:
     Very Low       Low  Somewhat Low  Somewhat High  High  Very High
TPR  0.090444  0.104762      0.840909       0.308688   0.0   0.013158
TNR  0.992793  0.988908      0.267301       0.871795   1.0   1.000000
FPR  0.007207  0.011092      0.732699       0.128205   0.0   0.000000
FNR  0.909556  0.895238      0.159091       0.691312   1.0   0.986842
```



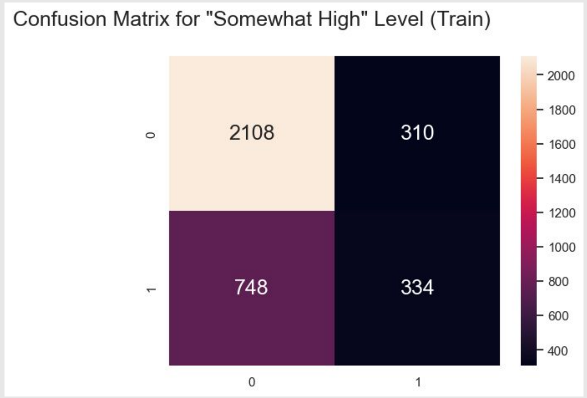


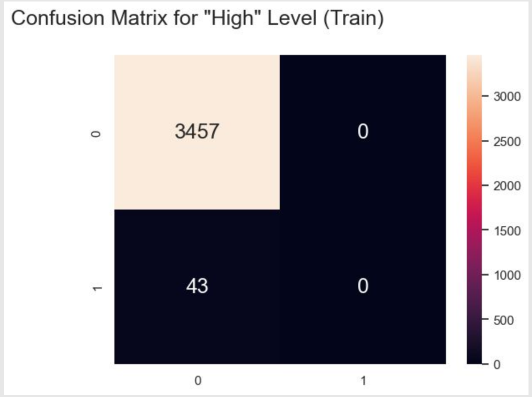


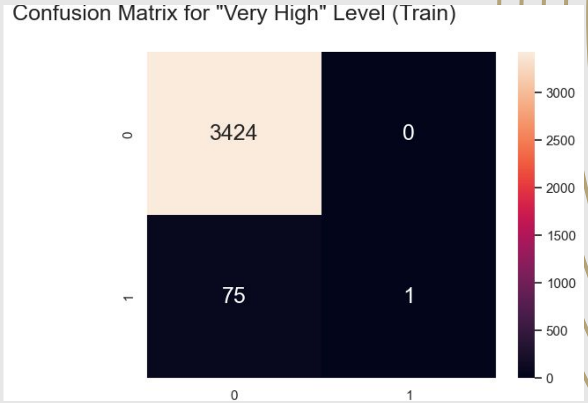
Check the accuracy of dectree_1



```
Train Data
Accuracy  :      0.412

Train Accuracy and Error rates:
     Very Low       Low  Somewhat Low  Somewhat High  High  Very High
TPR  0.090444  0.104762      0.840909       0.308688   0.0   0.013158
TNR  0.992793  0.988908      0.267301       0.871795   1.0   1.000000
FPR  0.007207  0.011092      0.732699       0.128205   0.0   0.000000
FNR  0.909556  0.895238      0.159091       0.691312   1.0   0.986842
```







Reflection and Refinement

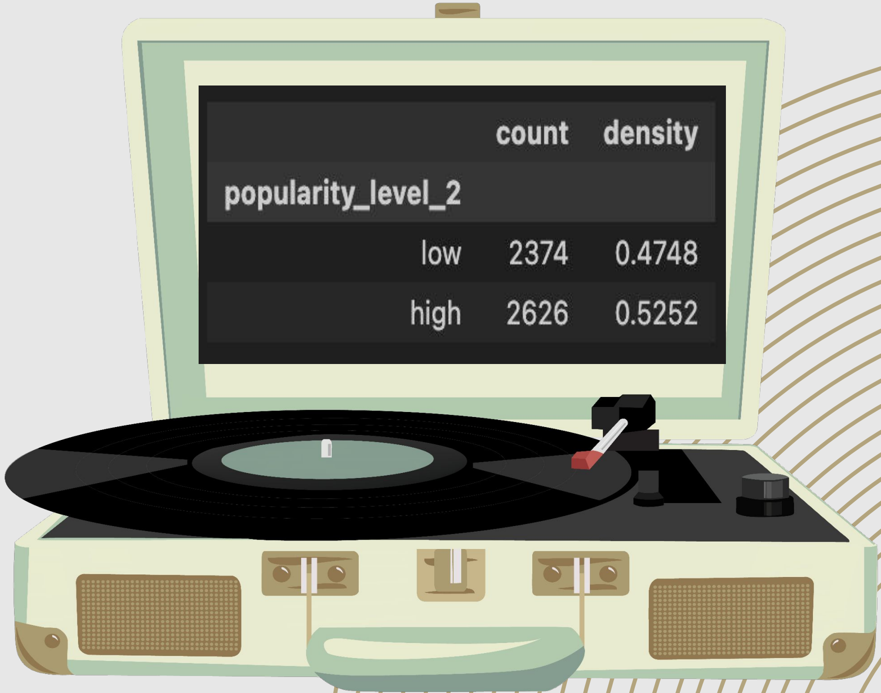
Current issue:

$$\begin{tikzcd} \text{TNR: high} & \text{TPR: low} \arrow[d] \\ & \text{FNR: high} \end{tikzcd}$$

Solution:

number of categories in `popularity_levels` from 6 to 2: `high` and `low`.

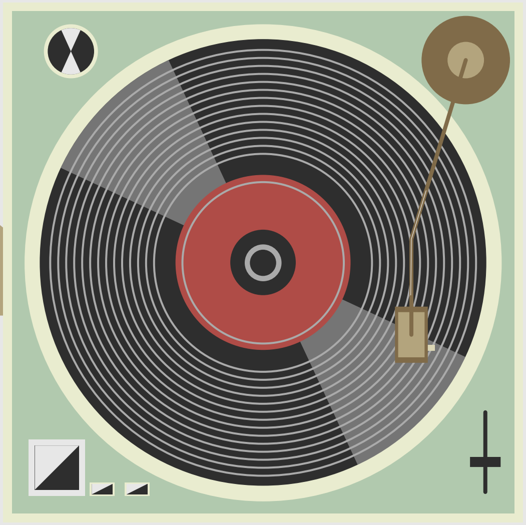
- high: (`track_popularity >= mean`)
- low: (`track_popularity < mean`)



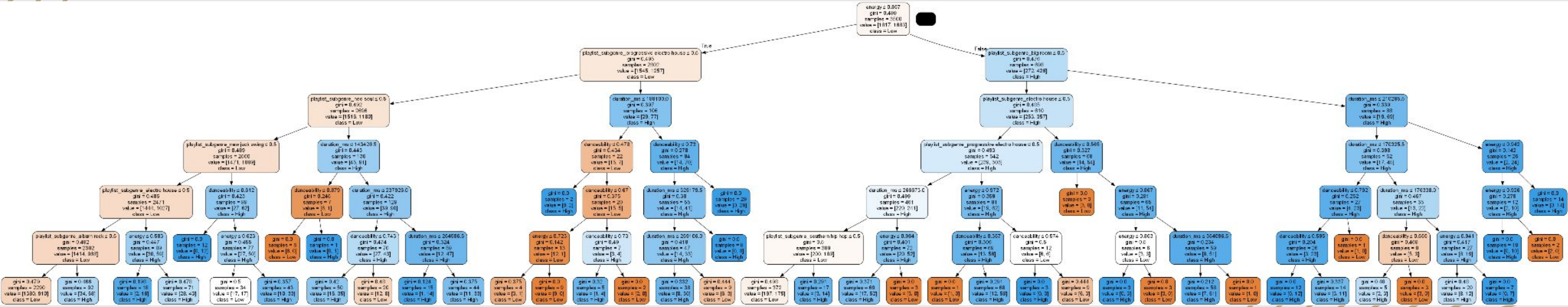
	count	density
popularity_level_2		
low	2374	0.4748
high	2626	0.5252

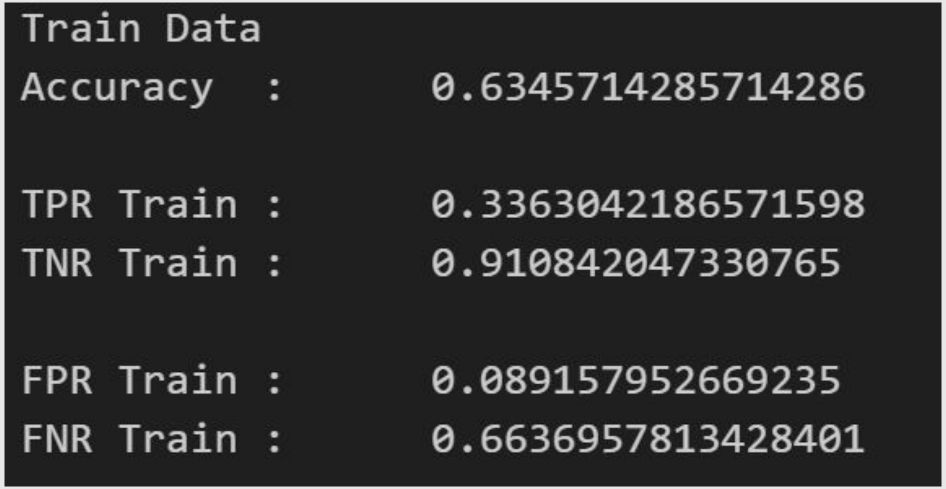
Model 2 & 3

Decision Tree Classification & Random Forest Classification

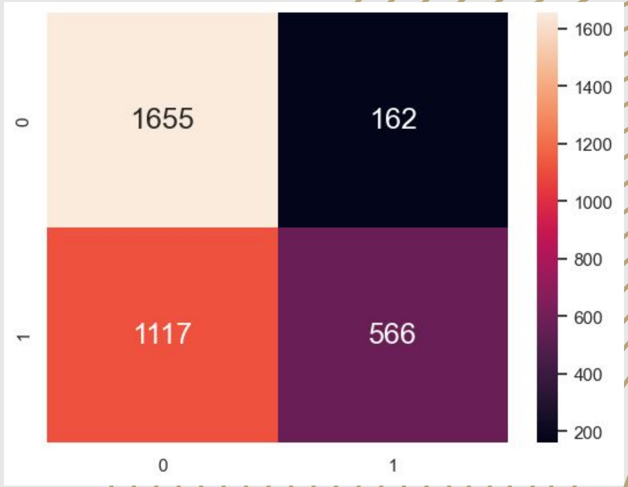


Accuracy of dectree_2 (Train)

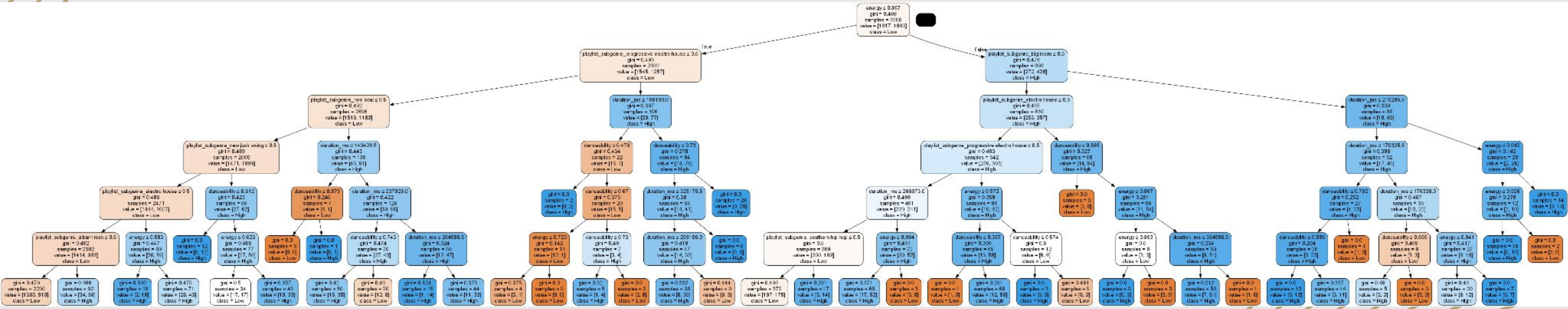


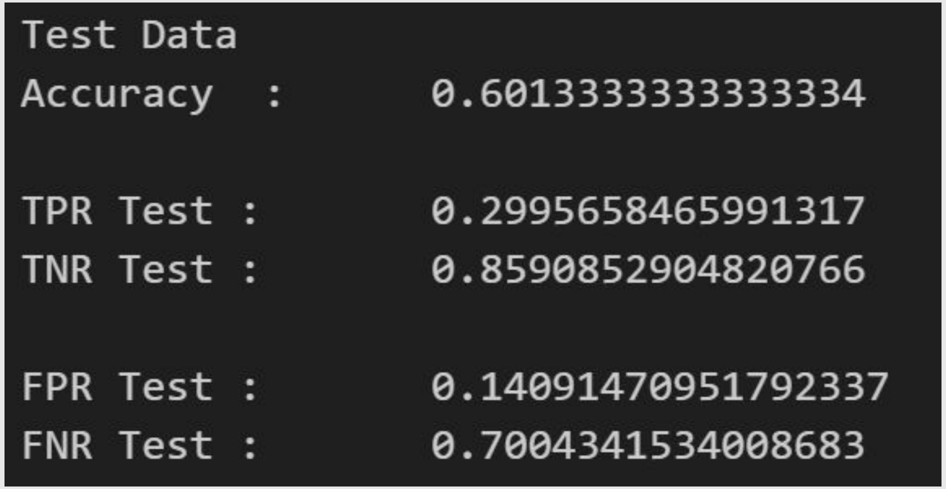


```
Train Data
Accuracy  :       0.6345714285714286

TPR Train :       0.3363042186571598
TNR Train :       0.910842047330765

FPR Train :       0.089157952669235
FNR Train :       0.6636957813428401
```

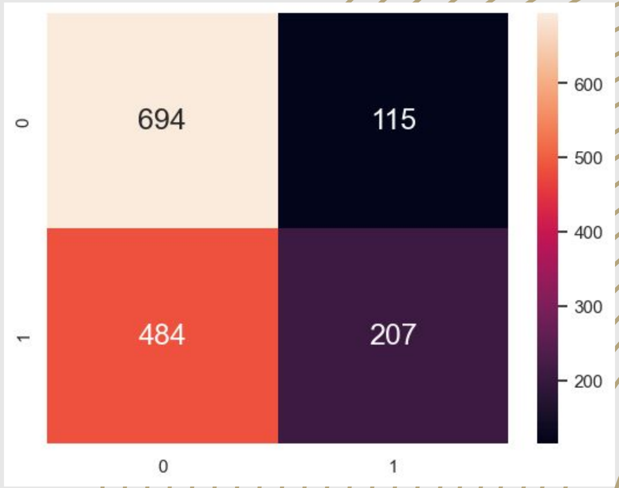


Accuracy of dectree_2 (Test)





```
Test Data
Accuracy  :      0.6013333333333334

TPR Test :       0.2995658465991317
TNR Test :       0.8590852904820766

FPR Test :       0.14091470951792337
FNR Test :       0.7004341534008683
```



Attempt to Predict using Random Forest

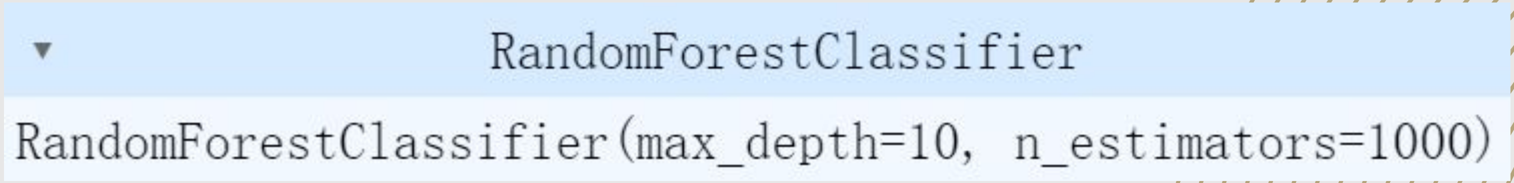
Hyperparameters:

n_estimators: **1000**

The number of trees in the forest

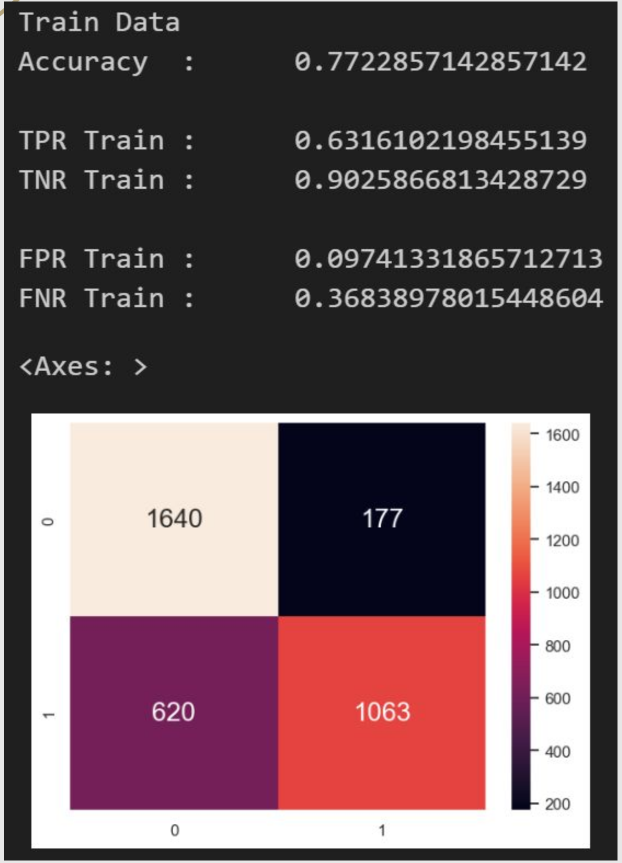
max_depth: **10**

The maximum depth of each tree

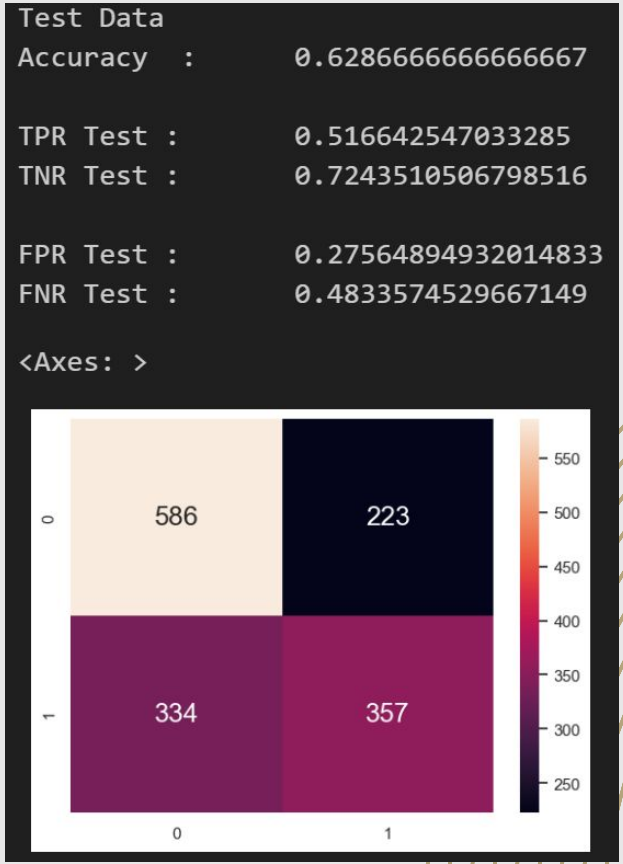


```
▾ RandomForestClassifier
RandomForestClassifier(max_depth=10, n_estimators=1000)
```

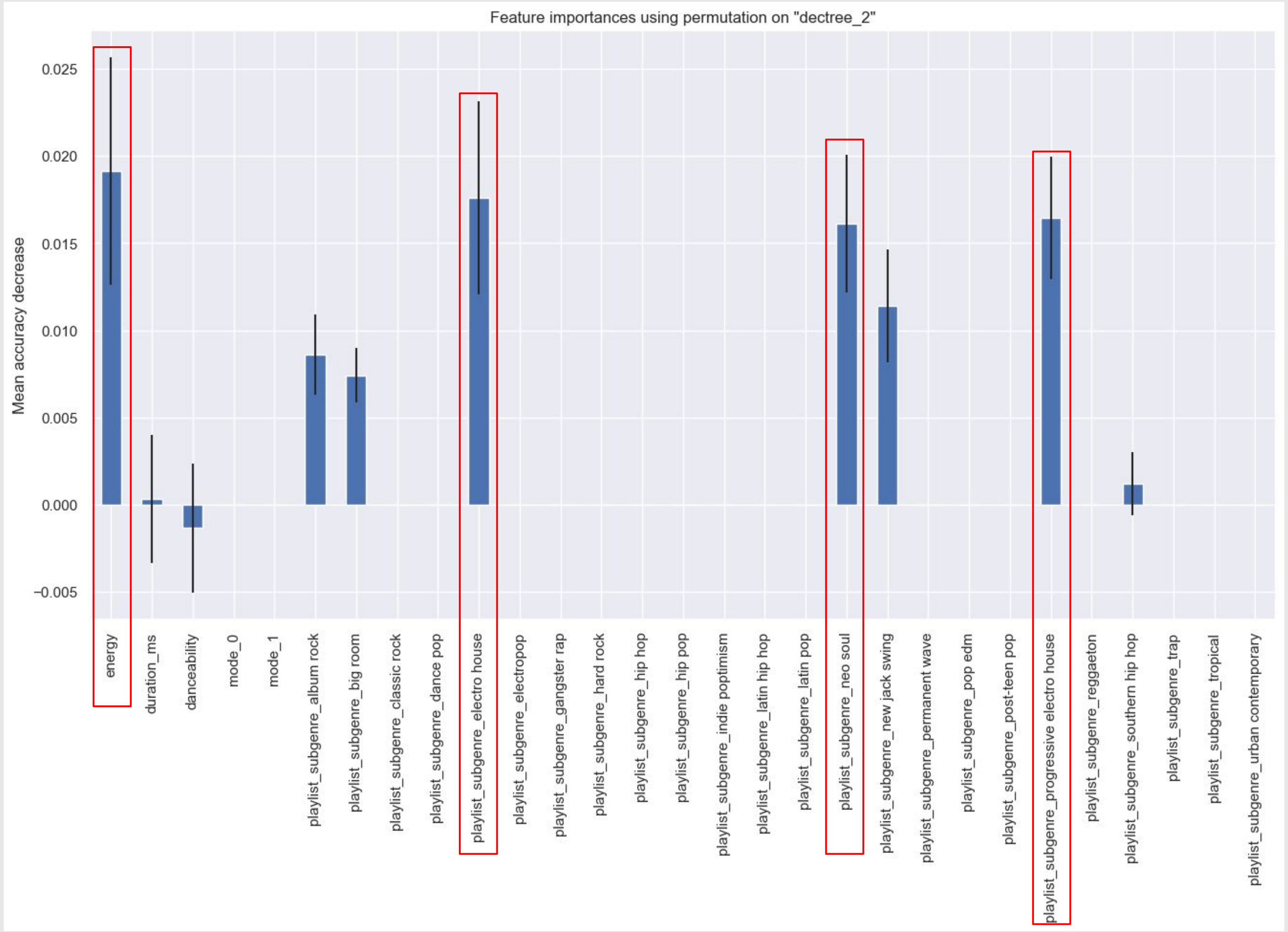
Accuracy of rforest

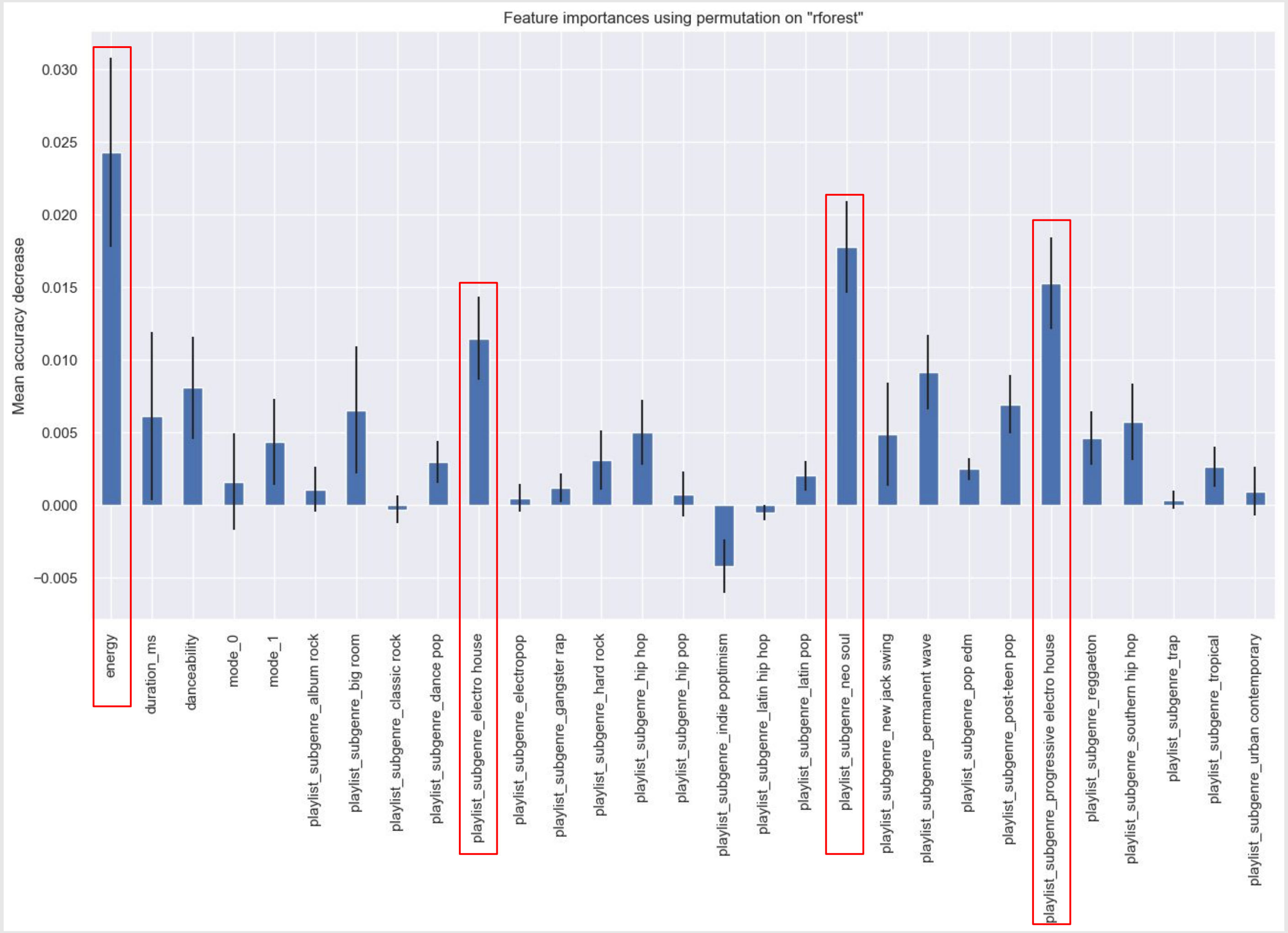


```
Train Data
Accuracy  :        0.7722857142857142

TPR Train :        0.6316102198455139
TNR Train :        0.9025866813428729

FPR Train :        0.09741331865712713
FNR Train :        0.36838978015448604

<Axes: >
```



```
Test Data
Accuracy  :        0.6286666666666667

TPR Test :         0.516642547033285
TNR Test :         0.7243510506798516

FPR Test :         0.27564894932014833
FNR Test :         0.4833574529667149

<Axes: >
```


Feature importances using permutation on "dectree_2"
Mean accuracy decrease
0.025
0.020
0.015
0.010
0.005
0.000
−0.005
energy
duration_ms
danceability
mode_0
mode_1
playlist_subgenre_album rock
playlist_subgenre_big room
playlist_subgenre_classic rock
playlist_subgenre_dance pop
playlist_subgenre_electro house
playlist_subgenre_electropop
playlist_subgenre_gangster rap
playlist_subgenre_hard rock
playlist_subgenre_hip hop
playlist_subgenre_hip pop
playlist_subgenre_indie poptimism
playlist_subgenre_latin hip hop
playlist_subgenre_latin pop
playlist_subgenre_neo soul
playlist_subgenre_new jack swing
playlist_subgenre_permanent wave
playlist_subgenre_pop edm
playlist_subgenre_post-teen pop
playlist_subgenre_progressive electro house
playlist_subgenre_reggaeton
playlist_subgenre_southern hip hop
playlist_subgenre_trap
playlist_subgenre_tropical
playlist_subgenre_urban contemporary


Feature importances using permutation on "rforest"
Mean accuracy decrease
0.030
0.025
0.020
0.015
0.010
0.005
0.000
−0.005
energy
duration_ms
danceability
mode_0
mode_1
playlist_subgenre_album rock
playlist_subgenre_big room
playlist_subgenre_classic rock
playlist_subgenre_dance pop
playlist_subgenre_electro house
playlist_subgenre_electropop
playlist_subgenre_gangster rap
playlist_subgenre_hard rock
playlist_subgenre_hip hop
playlist_subgenre_hip pop
playlist_subgenre_indie poptimism
playlist_subgenre_latin hip hop
playlist_subgenre_latin pop
playlist_subgenre_neo soul
playlist_subgenre_new jack swing
playlist_subgenre_permanent wave
playlist_subgenre_pop edm
playlist_subgenre_post-teen pop
playlist_subgenre_progressive electro house
playlist_subgenre_reggaeton
playlist_subgenre_southern hip hop
playlist_subgenre_trap
playlist_subgenre_tropical
playlist_subgenre_urban contemporary

Conclusion

- Spotify users prefer the more ***energized*** tracks
- Spotify users tend to appreciate the ***unconventional expression of emotions***

Thanks