COMP-4448

Christopher Kramer

2021-01-04

Spotify energy score via KNN

In [1]:
```python
import pandas as pd
import numpy as np
from itertools import repeat
from typing import Union
import seaborn as sns
from collections import Counter
```

Find your own dataset suitable for classification or regression with at least three input variables and 200 or more cases: Depending on the target variable of interest, you would build a k-nearest neighbor classifier or regressor using the appropriate sklearn estimator. Find some interesting unique dataset that is not popularly used in the internet. Address the following and include code/output snippets from b) to f). Include the response under each sub question.

a) State your research question

Can Spotify's "energy" score (user-generated aggregate metric) be predicted from Spotify song metadata using KNN regression?

Data from: https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

b) Data pre-processing (to the extent deemed necessary: remember the knn algorithm depends on distances, so you need to rescale, normalize or standardize your input values to make sure no variable influences the predictions due to it scale).

In [56]:
```python
spotify = pd.read_csv('data.csv')
```

In [57]:
```python
spotify.sample(5)
```

Out[57]:

	acousticness	artists	danceability	duration_ms	energy	explicit	id	instrumentalness	key	liveness
18674	0.2200	['Ty Dolla $ign', 'The Weeknd', 'Wiz Khalifa',...	0.805	242983	0.330	1	7t2bFihaDvhIrd2gn2CWJO	0.000000	1	0.1050
29587	0.0574	['Rod Stewart']	0.546	359733	0.859	0	7cYuolexJpsjD91G6i7XIm	0.000000	7	0.0855
15551	0.4220	['Billy Bragg', 'Wilco']	0.622	298533	0.736	0	38paDDziQ57k1f4VfKTeGk	0.000012	9	0.0829
45074	0.6290	['Richard Strauss', 'Fritz Reiner', 'Chicago S...	0.199	93587	0.207	0	2BdJKDe828AtUAEuZa50ZT	0.793000	5	0.0634
120806	0.5640	['Queen']	0.638	262133	0.576	0	0sA5xCFx2bF3jrz5Y5r0m1	0.000006	9	0.1560

Check nulls

In [58]:
```python
spotify.isna().sum()
```

Out[58]:
```
acousticness        0
artists             0
danceability        0
duration_ms         0
energy              0
explicit            0
id                  0
instrumentalness    0
key                 0
liveness            0
loudness            0
mode                0
name                0
popularity          0
release_date        0
speechiness         0
tempo               0
valence             0
year                0
dtype: int64
```

Drop string columns

In [59]:
```python
spotify.dtypes
```

Out[59]:
```
acousticness        float64
artists              object
danceability        float64
duration_ms           int64
energy              float64
explicit              int64
id                   object
instrumentalness    float64
key                   int64
liveness            float64
loudness            float64
mode                  int64
name                 object
popularity            int64
release_date         object
speechiness         float64
tempo               float64
valence             float64
year                  int64
dtype: object
```

In [60]:
```python
spotify = spotify.select_dtypes(['float64', 'int64'])
```

In [61]:
```python
spotify.dtypes
```

Out[61]:
```
acousticness        float64
danceability        float64
duration_ms           int64
energy              float64
explicit              int64
instrumentalness    float64
key                   int64
liveness            float64
loudness            float64
mode                  int64
popularity            int64
speechiness         float64
tempo               float64
valence             float64
year                  int64
dtype: object
```

Baseline 'year' column

In [62]:
```python
spotify['year'] = pd.Timestamp.now().year - spotify['year']
```

Get dummies

In [63]:
```python
spotify = pd.get_dummies(spotify, columns = ['explicit', 'key', 'mode'])
```

In [64]:
```python
spotify.dtypes
```

Out[64]:
```
acousticness        float64
danceability        float64
duration_ms           int64
energy              float64
instrumentalness    float64
liveness            float64
loudness            float64
popularity            int64
speechiness         float64
tempo               float64
valence             float64
year                  int64
explicit_0            uint8
explicit_1            uint8
key_0                 uint8
key_1                 uint8
key_2                 uint8
key_3                 uint8
key_4                 uint8
key_5                 uint8
key_6                 uint8
key_7                 uint8
key_8                 uint8
key_9                 uint8
key_10                uint8
key_11                uint8
mode_0                uint8
mode_1                uint8
dtype: object
```

Scale Dummies

In [65]:
```python
spotify[spotify.select_dtypes('uint8').columns] = spotify[spotify.select_dtypes('uint8').columns].replace({0:-1})
```

Scale Floats

In [66]:
```python
from sklearn.compose import make_column_transformer, make_column_selector
from sklearn.preprocessing import MinMaxScaler
```

In [67]:
```python
X = spotify[spotify.columns[~spotify.columns.isin(['energy'])]]
y = spotify['energy']
```

In [68]:
```python
X
```

Out[68]:

	acousticness	danceability	duration_ms	instrumentalness	liveness	loudness	popularity	speechiness	tempo	valence	...	ke
0	0.991000	0.598	168333	0.000522	0.3790	-12.628	12	0.0936	149.976	0.6340	...	
1	0.643000	0.852	150200	0.026400	0.0809	-7.261	7	0.0534	86.889	0.9500	...	
2	0.993000	0.647	163827	0.000018	0.5190	-12.098	4	0.1740	97.600	0.6890	...	
3	0.000173	0.730	422087	0.801000	0.1280	-7.311	17	0.0425	127.997	0.0422	...	
4	0.295000	0.704	165224	0.000246	0.4020	-6.036	2	0.0768	122.076	0.2990	...	
...	...	...	...	...	...	...	...	...	...	...	...	
174384	0.009170	0.792	147615	0.000060	0.1780	-5.089	0	0.0356	125.972	0.1860	...	
174385	0.795000	0.429	144720	0.000000	0.1960	-11.665	0	0.0360	94.710	0.2280	...	
174386	0.806000	0.671	218147	0.920000	0.1130	-12.393	0	0.0282	108.058	0.7140	...	
174387	0.920000	0.462	244000	0.000000	0.1130	-12.077	69	0.0377	171.319	0.3200	...	
174388	0.239000	0.677	197710	0.891000	0.2150	-12.237	0	0.0258	112.208	0.7470	...	

174389 rows × 27 columns

In [69]:
```python
ct = make_column_transformer((StandardScaler(), X.columns[:list(X.columns).index('year')+1]),remainder='passthrough')
```

In [71]:
```python
X = pd.DataFrame(ct.fit_transform(X), columns=X.columns)
X
```

Out[71]:

	acousticness	danceability	duration_ms	instrumentalness	liveness	loudness	popularity	speechiness	tempo	valence	..
0	1.294358	0.347919	-0.434495	-0.588004	0.930106	-0.154111	-0.626050	-0.066549	1.089753	0.413903	..
1	0.378411	1.790898	-0.556689	-0.510657	-0.721489	0.788862	-0.854645	-0.287113	-0.995485	1.608718	..
2	1.299622	0.626289	-0.464860	-0.589511	1.705763	-0.060991	-0.991803	0.374580	-0.641450	0.621861	..
3	-1.313529	1.097814	1.275491	1.804534	-0.460536	0.780077	-0.397454	-0.346918	0.363273	-1.823729	..
4	-0.537536	0.950107	-0.455446	-0.588829	1.057535	1.004092	-1.083241	-0.158725	0.167564	-0.852753	..
...	...	...	...	...	...	...	...	...	...	...	..
174384	-1.289849	1.450037	-0.574108	-0.589385	-0.183516	1.170478	-1.174679	-0.384776	0.296340	-1.280013	..
174385	0.778480	-0.612173	-0.593617	-0.589564	-0.083788	0.015086	-1.174679	-0.382582	-0.736975	-1.121208	..
174386	0.807432	0.762634	-0.098811	2.160212	-0.543642	-0.112822	-1.174679	-0.425378	-0.295778	0.716387	..
174387	1.107484	-0.424699	0.075406	-0.589564	-0.543642	-0.057301	1.979941	-0.373254	1.795212	-0.773351	..
174388	-0.684929	0.796720	-0.236531	2.073534	0.021479	-0.085413	-1.174679	-0.438546	-0.158607	0.841162	..

174389 rows × 27 columns

c) Data splitting

In [72]:
```python
from sklearn.model_selection import train_test_split
```

In [73]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=42)
```

In [74]:
```python
X_train.shape, y_train.shape
```

Out[74]:
```
((139511, 27), (139511,))
```

In [75]:
```python
X_test.shape, y_test.shape
```

Out[75]:
```
((34878, 27), (34878,))
```

d) Model construction

In [76]:
```python
knn = KNeighborsRegressor(n_jobs=-1)
```

In [77]:
```python
knn.fit(X_train, y_train)
```

Out[77]:
```
KNeighborsRegressor(n_jobs=-1)
```

Base metrics

In [78]:
```python
from sklearn.metrics import r2_score
```

In [79]:
```python
y_pred_train = knn.predict(X_train)
```

In [80]:
```python
y_pred_test = knn.predict(X_test)
```

Train

In [81]:
```python
r2_score(y_train, y_pred_train)
```

Out[81]:
```
0.8692657365579515
```

In [82]:
```python
mean_squared_error(y_train, y_pred_train, squared=False)
```

Out[82]:
```
0.09855464044834011
```

Test

In [83]:
```python
r2_score(y_test, y_pred_test)
```

Out[83]:
```
0.8029904002733431
```

In [84]:
```python
mean_squared_error(y_test, y_pred_test, squared=False)
```

Out[84]:
```
0.12123105702934114
```

e) Hyperparameter turning (choose whatever approach your like)

In [ ]:
```python
# trying tune-sklearn which is supposed to be faster than built-in tuners.
from tune_sklearn import TuneSearchCV
```

In [88]:
```python
# This dataset takes a long time to train, so I've limited my choices and number of folds
params = {
    'n_neighbors': list(range(1, 52, 10))
}
```

In [96]:
```python
# Below can be replaced with "GridSearchCV" for classic tuning
grid = TuneSearchCV(knn, params, n_jobs=10, verbose=2, cv=4, use_gpu = True)
```

In [97]:
```python
grid.fit(X_train, y_train)
```

```
Checkpointing the experiment state took 1.796 s, which may be a performance bottleneck. Please ensure the `TUNE_GLOBAL_CHECKPOINT_S` environment variable is something significantly higher than this duration to ensure compute time is mostly spent on the main training loop.
```

== Status ==
Memory usage on this node: 10.8/31.9 GiB
Using FIFO scheduling algorithm.
Resources requested: 2/16 CPUs, 0.1/1 GPUs, 0.0/13.57 GiB heap, 0.0/4.64 GiB objects
Result logdir: C:\Users\KittheKat\ray_results_Trainable_2021-02-15_17-28-14
Number of trials: 1/5 (1 RUNNING)

```
The `start_trial` operation took 0.643 s, which may be a performance bottleneck.
```

```
Trial _Trainable_db558_00000 reported split0_test_score=0.8070193020045853,split1_test_score=0.8047274625036641,split2_test_score=0.8035835462868018,split3_test_score=0.8030203857801017,average_test_score=0.8045876741437882,objective=0.8045876741437882 with parameters={'early_stopping': False, 'early_stop_type': <EarlyStopping.NO_EARLY_STOP: 7>, 'X_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000008000000), 'y_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000009000000), 'groups': None, 'cv': KFold(n_splits=4, random_state=None, shuffle=False), 'fit_params': {}, 'scoring': {'score': <function _passthrough_scorer at 0x000002986C6F50D0>}, 'max_iters': 1, 'return_train_score': False, 'n_jobs': 1, 'metric_name': 'average_test_score', 'n_neighbors': 11, 'estimator_list': [KNeighborsRegressor(n_jobs=-1)]}. This trial completed.
```

```
The `process_trial_result` operation took 1.558 s, which may be a performance bottleneck.
Processing trial results took 1.560 s, which may be a performance bottleneck. Please consider reporting results less frequently to Ray Tune.
The `process_trial` operation took 1.561 s, which may be a performance bottleneck.
Checkpointing the experiment state took 113.195 s, which may be a performance bottleneck. Please ensure the `TUNE_GLOBAL_CHECKPOINT_S` environment variable is something significantly higher than this duration to ensure compute time is mostly spent on the main training loop.
Trial Runner checkpointing failed: [WinError 183] Cannot create a file when that file already exists: 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\.tmp_generator' -> 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\basic-variant-state-2021-02-15_17-28-14.json'
```

== Status ==
Memory usage on this node: 17.1/31.9 GiB
Using FIFO scheduling algorithm.
Resources requested: 8/16 CPUs, 0.4/1 GPUs, 0.0/13.57 GiB heap, 0.0/4.64 GiB objects
Result logdir: C:\Users\KittheKat\ray_results_Trainable_2021-02-15_17-28-14
Number of trials: 5/5 (4 RUNNING, 1 TERMINATED)

```
The `on_step_begin` operation took 0.945 s, which may be a performance bottleneck.
The `get_next_failed_trial` operation took 0.955 s, which may be a performance bottleneck.
```

```
Trial _Trainable_db558_00002 reported split0_test_score=0.8047373708677723,split1_test_score=0.8022934609645567,split2_test_score=0.8026419798556756,split3_test_score=0.8011765557721433,average_test_score=0.8027123418650369,objective=0.8027123418650369 with parameters={'early_stopping': False, 'early_stop_type': <EarlyStopping.NO_EARLY_STOP: 7>, 'X_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000008000000), 'y_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000009000000), 'groups': None, 'cv': KFold(n_splits=4, random_state=None, shuffle=False), 'fit_params': {}, 'scoring': {'score': <function _passthrough_scorer at 0x000002986C6F50D0>}, 'max_iters': 1, 'return_train_score': False, 'n_jobs': 1, 'metric_name': 'average_test_score', 'n_neighbors': 21, 'estimator_list': [KNeighborsRegressor(n_jobs=-1)]}. This trial completed.
```

```
Checkpointing the experiment state took 15.020 s, which may be a performance bottleneck. Please ensure the `TUNE_GLOBAL_CHECKPOINT_S` environment variable is something significantly higher than this duration to ensure compute time is mostly spent on the main training loop.
Trial Runner checkpointing failed: [WinError 183] Cannot create a file when that file already exists: 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\.tmp_generator' -> 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\basic-variant-state-2021-02-15_17-28-14.json'
```

== Status ==
Memory usage on this node: 16.3/31.9 GiB
Using FIFO scheduling algorithm.
Resources requested: 6/16 CPUs, 0.3/1 GPUs, 0.0/13.57 GiB heap, 0.0/4.64 GiB objects
Result logdir: C:\Users\KittheKat\ray_results_Trainable_2021-02-15_17-28-14
Number of trials: 5/5 (3 RUNNING, 2 TERMINATED)

```
Trial _Trainable_db558_00003 reported split0_test_score=0.798619731128189,split1_test_score=0.7962843298862314,split2_test_score=0.7959358894703314,split3_test_score=0.7953211430923094,average_test_score=0.7965402733942654,objective=0.7965402733942654 with parameters={'early_stopping': False, 'early_stop_type': <EarlyStopping.NO_EARLY_STOP: 7>, 'X_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000008000000), 'y_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000009000000), 'groups': None, 'cv': KFold(n_splits=4, random_state=None, shuffle=False), 'fit_params': {}, 'scoring': {'score': <function _passthrough_scorer at 0x000002986C6F50D0>}, 'max_iters': 1, 'return_train_score': False, 'n_jobs': 1, 'metric_name': 'average_test_score', 'n_neighbors': 41, 'estimator_list': [KNeighborsRegressor(n_jobs=-1)]}. This trial completed.
```

```
Checkpointing the experiment state took 11.359 s, which may be a performance bottleneck. Please ensure the `TUNE_GLOBAL_CHECKPOINT_S` environment variable is something significantly higher than this duration to ensure compute time is mostly spent on the main training loop.
Trial Runner checkpointing failed: [WinError 183] Cannot create a file when that file already exists: 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\.tmp_generator' -> 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\basic-variant-state-2021-02-15_17-28-14.json'
```

== Status ==
Memory usage on this node: 15.3/31.9 GiB
Using FIFO scheduling algorithm.
Resources requested: 4/16 CPUs, 0.2/1 GPUs, 0.0/13.57 GiB heap, 0.0/4.64 GiB objects
Result logdir: C:\Users\KittheKat\ray_results_Trainable_2021-02-15_17-28-14
Number of trials: 5/5 (2 RUNNING, 3 TERMINATED)

```
Trial _Trainable_db558_00004 reported split0_test_score=0.8015592205895563,split1_test_score=0.7995203593510085,split2_test_score=0.7990281333235592,split3_test_score=0.7981037617861465,average_test_score=0.7995528687625676,objective=0.7995528687625676 with parameters={'early_stopping': False, 'early_stop_type': <EarlyStopping.NO_EARLY_STOP: 7>, 'X_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000008000000), 'y_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000009000000), 'groups': None, 'cv': KFold(n_splits=4, random_state=None, shuffle=False), 'fit_params': {}, 'scoring': {'score': <function _passthrough_scorer at 0x000002986C6F50D0>}, 'max_iters': 1, 'return_train_score': False, 'n_jobs': 1, 'metric_name': 'average_test_score', 'n_neighbors': 31, 'estimator_list': [KNeighborsRegressor(n_jobs=-1)]}. This trial completed.
```

```
Checkpointing the experiment state took 12.651 s, which may be a performance bottleneck. Please ensure the `TUNE_GLOBAL_CHECKPOINT_S` environment variable is something significantly higher than this duration to ensure compute time is mostly spent on the main training loop.
Trial Runner checkpointing failed: [WinError 183] Cannot create a file when that file already exists: 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\.tmp_generator' -> 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\basic-variant-state-2021-02-15_17-28-14.json'
```

== Status ==
Memory usage on this node: 14.4/31.9 GiB
Using FIFO scheduling algorithm.
Resources requested: 2/16 CPUs, 0.1/1 GPUs, 0.0/13.57 GiB heap, 0.0/4.64 GiB objects
Result logdir: C:\Users\KittheKat\ray_results_Trainable_2021-02-15_17-28-14
Number of trials: 5/5 (1 RUNNING, 4 TERMINATED)

```
Trial _Trainable_db558_00001 reported split0_test_score=0.798619731128189,split1_test_score=0.7962843298862314,split2_test_score=0.7959358894703314,split3_test_score=0.7953211430923094,average_test_score=0.7965402733942654,objective=0.7965402733942654 with parameters={'early_stopping': False, 'early_stop_type': <EarlyStopping.NO_EARLY_STOP: 7>, 'X_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000008000000), 'y_id': ObjectRef(ffffffffffffffffffffffffffffffffffffffff0100000009000000), 'groups': None, 'cv': KFold(n_splits=4, random_state=None, shuffle=False), 'fit_params': {}, 'scoring': {'score': <function _passthrough_scorer at 0x000002986C6F50D0>}, 'max_iters': 1, 'return_train_score': False, 'n_jobs': 1, 'metric_name': 'average_test_score', 'n_neighbors': 41, 'estimator_list': [KNeighborsRegressor(n_jobs=-1)]}. This trial completed.
```

```
Checkpointing the experiment state took 13.801 s, which may be a performance bottleneck. Please ensure the `TUNE_GLOBAL_CHECKPOINT_S` environment variable is something significantly higher than this duration to ensure compute time is mostly spent on the main training loop.
Trial Runner checkpointing failed: [WinError 183] Cannot create a file when that file already exists: 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\.tmp_generator' -> 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\basic-variant-state-2021-02-15_17-28-14.json'
```

== Status ==
Memory usage on this node: 13.4/31.9 GiB
Using FIFO scheduling algorithm.
Resources requested: 0/16 CPUs, 0.0/1 GPUs, 0.0/13.57 GiB heap, 0.0/4.64 GiB objects
Result logdir: C:\Users\KittheKat\ray_results_Trainable_2021-02-15_17-28-14
Number of trials: 5/5 (5 TERMINATED)

```
Trial Runner checkpointing failed: [WinError 183] Cannot create a file when that file already exists: 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\.tmp_generator' -> 'C:\\Users\\KittheKat\\ray_results\\_Trainable_2021-02-15_17-28-14\\basic-variant-state-2021-02-15_17-28-14.json'
```

== Status ==
Memory usage on this node: 13.4/31.9 GiB
Using FIFO scheduling algorithm.
Resources requested: 0/16 CPUs, 0.0/1 GPUs, 0.0/13.57 GiB heap, 0.0/4.64 GiB objects
Result logdir: C:\Users\KittheKat\ray_results_Trainable_2021-02-15_17-28-14
Number of trials: 5/5 (5 TERMINATED)

Trial name	status	loc	n_neighbors	iter	total time (s)	split0_test_score	split1_test_score	split2_test_score
_Trainable_db558_00000	TERMINATED		11	1	301.131	0.807019	0.804727	0.803584
_Trainable_db558_00001	TERMINATED		41	1	456.843	0.79862	0.796284	0.795936
_Trainable_db558_00002	TERMINATED		21	1	443.254	0.804737	0.802293	0.802642
_Trainable_db558_00003	TERMINATED		41	1	452.329	0.79862	0.796284	0.795936
_Trainable_db558_00004	TERMINATED		31	1	453.878	0.801559	0.79952	0.799028

```
(pid=24860) Windows fatal exception: access violation
(pid=24860)
(pid=24860) Windows fatal exception: access violation
(pid=24860)
```

Out[97]:
```
TuneSearchCV(cv=4, estimator=KNeighborsRegressor(n_jobs=-1),
             loggers=[<class 'ray.tune.logger.CSVLogger'>,
                      <class 'ray.tune.logger.JsonLogger'>],
             n_jobs=10, n_trials=5,
             param_distributions={'n_neighbors': [1, 11, 21, 31, 41]},
             scoring={'score': <function _passthrough_scorer at 0x000002986C6F50D0>},
             sk_n_jobs=1, use_gpu=True, verbose=2)
```

In [94]:
```python
import joblib
```

In [95]:
```python
joblib.dump(grid, 'grid_search.joblib')
```

Out[95]:
```
['grid_search.joblib']
```

In [98]:
```python
grid.best_params_
```

Out[98]:
```
{'n_neighbors': 11}
```

Micro-tuning

In [111]:
```python
# This dataset takes a long time to train, so I've limited my choices and number of folds
params = {
    'n_neighbors': [11, 12, 13, 14, 15, 16]
}
```

In [112]:
```python
microgrid = GridSearchCV(KNeighborsRegressor(), params, n_jobs=-1, verbose=2, cv=4)
```

In [113]:
```python
microgrid.fit(X_train, y_train)
```

```
Fitting 4 folds for each of 6 candidates, totalling 24 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done   6 out of  24 | elapsed:  5.3min remaining: 16.0min
[Parallel(n_jobs=-1)]: Done  19 out of  24 | elapsed:  8.6min remaining:  2.3min
[Parallel(n_jobs=-1)]: Done  24 out of  24 | elapsed:  8.8min finished
```

Out[113]:
```
GridSearchCV(cv=4, estimator=KNeighborsRegressor(), n_jobs=-1,
             param_grid={'n_neighbors': [11, 12, 13, 14, 15, 16]}, verbose=2)
```

In [114]:
```python
joblib.dump(microgrid, 'microgrid_search.joblib')
```

Out[114]:
```
['microgrid_search.joblib']
```

In [115]:
```python
microgrid.best_params_
```

Out[115]:
```
{'n_neighbors': 13}
```

f) Use the best or optimal parameter values to build a model, then compute the accuracy score for your estimator.

In [116]:
```python
knn_grid = microgrid.best_estimator_
```

In [117]:
```python
y_pred_train = knn_grid.predict(X_train)
y_pred_test = knn_grid.predict(X_test)
```

Training

In [118]:
```python
r2_score(y_train, y_pred_train)
```

Out[118]:
```
0.8380230790180042
```

In [119]:
```python
mean_squared_error(y_train, y_pred_train, squared=False)
```

Out[119]:
```
0.10970058290931115
```

Test

In [120]:
```python
r2_score(y_test, y_pred_test)
```

Out[120]:
```
0.8112129507922109
```

In [121]:
```python
mean_squared_error(y_test, y_pred_test, squared=False)
```

Out[121]:
```
0.11867419561831806
```

Discuss about overfitting for the model

Given the most recent (KNN Grid) test scores, I do not believe this model is overfit. While the training RMSE and R2 have gone down from the out-of-the-box model, the test scores have improved slightly. The gap between test and training scores is not enough significant at this point to make a determination that the model is overfit.

Overall, I was very impressed with this model. It is wildly computationally intensive, but I'm surprised that the model predicted so well using non-text (no artist name, etc.) features and without access to actual mp3/song data files.

In [ ]: