In [7]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
import copy
```

In [8]:

```
data_set = pd.read_csv(r'random_songs_processed_dataset.csv').dropna()
data_set.sample(3)
```

Out[8]:

	artist_name	track_id	track_name	acousticness	danceability	c
27948	Daron Malakian and Scars On Broadway	2SZF3qH7XF7205oNaeEs49	Guns Are Loaded	0.0254	0.532	
31487	Los Freddy's	0wksjZOuof6WX51ljnnav8	Es Mejor Que Te Olvidé	0.4300	0.797	
2424	Jorja Smith	14qLa09blyCJdkRJRQ8lpV	Tomorrow	0.6110	0.642	

3 rows × 282 columns

we removed some categorical features such as time_signature and key. We also tried using the artist's popularity as a feature, but the model's performance didn't improve significantly. Due to that, we decided to drop the artist's popularity and focus our research and conclusions only on the musical features.

In [9]:

```
features = data_set[['acousticness','danceability','energy','instrumentalness','livenes
s','mode','speechiness',
                     'valence','duration_norm','loudness_norm','tempo_norm']]
features.head()
```

Out[9]:

	acousticness	danceability	energy	instrumentalness	liveness	mode	speechiness	v
0	0.005820	0.743	0.339	0.000	0.0812	1	0.4090	
1	0.024400	0.846	0.557	0.000	0.2860	1	0.4570	
2	0.025000	0.603	0.723	0.000	0.0824	0	0.0454	
3	0.029400	0.800	0.579	0.912	0.0994	0	0.0701	
4	0.000035	0.783	0.792	0.878	0.0332	1	0.0661	

Linear Regression

let's start with a very simple model - linear regression. at first, we tried using the only the musical features for different countries (one example country from each cluster). we got very bad results(the optimal score for this model is 1, 0 is the score for a model returning the average popularity of the training set, below 0 meanns that the model was worst than the constant one). than we tried adding the artist popularity. As we can se bellow, this model still performs poorly

In [10]:

```
for country in ['United States','Australia','Germany','Pakistan','Chile','Jamaica', 'Ja
pan','popularity']:
    print('training model on ' + country)
    labels = data_set[country]
    train_features, test_features, train_labels, test_labels = train_test_split(feature
s, labels, test_size = 0.1)
    linear_regression = LinearRegression()
    linear_regression.fit(train_features,train_labels)
    print('score is: ' +str(linear_regression.score(test_features,test_labels)))
```

```
training model on United States
score is: 0.040806842688392186
training model on Australia
score is: 0.042346218980992
training model on Germany
score is: 0.019108897817751935
training model on Pakistan
score is: 0.017638365829087732
training model on Chile
score is: 0.016118821923736926
training model on Jamaica
score is: 0.021542036212948257
training model on Japan
score is: 0.005543550036014588
training model on popularity
score is: 0.07655551266552595
```

it makes sense that the connection between the different features and the song's popularity is not linear... let's try a more complex model, that can express more complicated relationships

Random Forest

even when using the artist popularity feature, and trying different hyper parameters, the random forest regression performs poorly...

In [11]:

```
for country in ['United States','Australia','Germany','Pakistan','Chile','Jamaica', 'Ja
pan','popularity']:
    print('training model on ' + country)
    labels = data_set[country]
    train_features, test_features, train_labels, test_labels = train_test_split(feature
s, labels, test_size = 0.1)
    random_forest = RandomForestRegressor(n_estimators=15,max_features ='sqrt')
    random_forest.fit(train_features,train_labels)
    print('score is: ' + str(random_forest.score(test_features,test_labels)))
```

```
training model on United States
score is: 0.043988583912793366
training model on Australia
score is: 0.016638729238479533
training model on Germany
score is: -0.06565958037228925
training model on Pakistan
score is: -0.08545911930447159
training model on Chile
score is: -0.026464761381213453
training model on Jamaica
score is: -0.03927793177037442
training model on Japan
score is: -0.0842034652089283
training model on popularity
score is: 0.06328677766508028
```