CISC 372 Advanced Data Analytics Final Project

Instructor: Steve Ding

Zihan Zhang

17zz55

20098936

Table of Contents

Abstract

In this paper, I assumed that music features will develop and have a linear relationship with the time passes and can be fit in a specific model that we could predict the year music producing according to the music features provided. However, after several trials and improvements, I finally found this idea may be false. The failure reasons maybe the size of the data or the sensibility of artwork which will be discussed in detail in this paper. In short, from the experiments, we found that music features do have a linear relationship with time but this relationship is not enough to support us build a model which could predict the year music producing.

Introduction

Nowadays, music products focus more and more on commercial profits. It may not be a good development for the music, but it is the reality every artist has to face. Music with only artistry gets harder and harder to survive in the capitalist society. So artists need to consider more about the trends among the audience. Try to produce some music with both artistry and popularity possibly could help them to achieve better investment and attention to support them chase a further dream. Overall, they need some data analysis on the big hits music and take advantage of the data to have a more detailed and clear acknowledgment of the popular trend.

How does the U.S. affection for popular songs change over time? Does the U.S. audience prefer more on hip-hop or electronic music along with the change of time? This time, we may have more data analysis on Billboard which has one of the world's most famous music charts. I would like to use linear regression to build the model, use the time as a dependent variable and the features of different music songs as the x_train set or the x_test set. Our assumption is that appearance of these types of songs will increase as time goes. Conversely, give some features of a popular song, it may be possible to predict its producing year. A linear transformation is kind of ideal result for that assumption.

The challenges we are facing is first, the data may not be sufficient. The data set only provides the best top songs in a year. The total number is 603 which steps across a decade. That means on average, there are only around 60 songs could be treat as training sample per year. Could this be enough for generating models to predict the year of producing? The second challenge is that probably the features offered in the data set is also insufficient, some features may even not be affected by the time flows.

However, once the model could be ideally generated, not only the music producer and their back companies but also the audience could be benefits from the model. Music producers could have a prediction on whether the work meets the trends these years. Furthermore, sometimes the music world needs some retro trends. The model could help the producer to know does the work satisfies the demand of retro trend. Companies could utilize the model to judge the invisible value behind works. The audience could assort music and choose the music on their appetites more conveniently with the help of the model.

Problem Statement

Separate the dataset into train set and test set by the year. Use the train set to generate the model and use that to give predictions. Then compare the predictions with the testing set to find out the accuracy of the model then judge that could this model be helpful to predict the producing year of pop music.

Proposed Method or Solution

GridsearchCV

Since the dataset is not diagrams or sequential data. Grid search method is a very simple method we learnt so far to fit the problem. We could try multiple times to find the best hyperparameters that generate the best model in our trials. Here I decide to treat 'bpm, nrgy, dnce, dB, live, val, acous, pop' and 'top genre' as my candidate hyperparameters. GridSearchCV will automatically apply the cross-validation to help present the best estimator.

Data Preprocessing

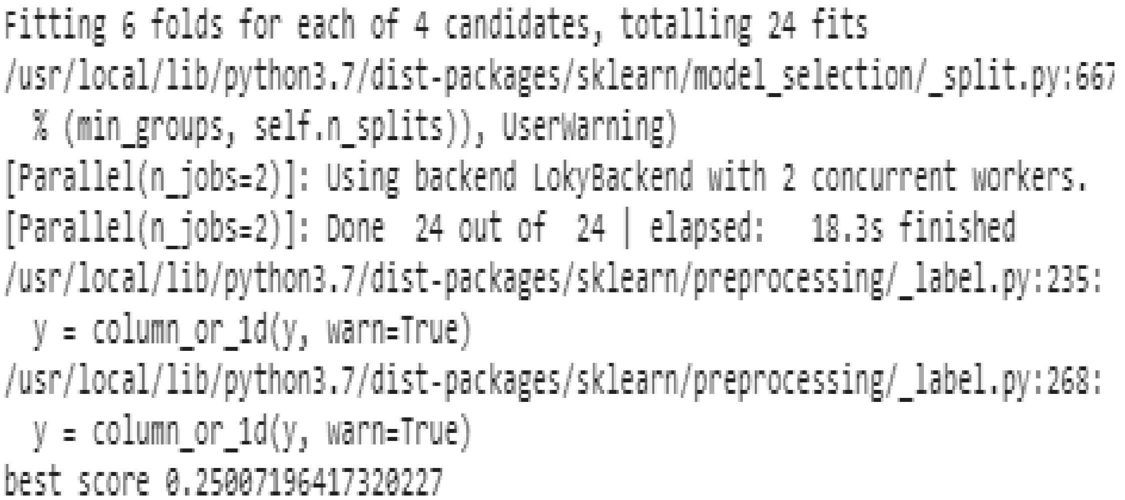
The data set provided in Kaggle seems to be well preprocessed. The strategy I choose to modify the data may not affect a lot. Just mention that I used the most frequent strategy for numeric data missing value filling, the constant strategy for categorical data missing value filling, and the OneHotEncoder to encode the features to numeric arrays.

Cohen_Kappa_Score

The default scoring strategy accuracy may be too strict for the year estimator. I think the float of year around one or two could be accepted. The kappa score will use the value of Kappa to make more assessments on the prediction which increases the value of accuracy of the model, but the predictions have not changed actually.

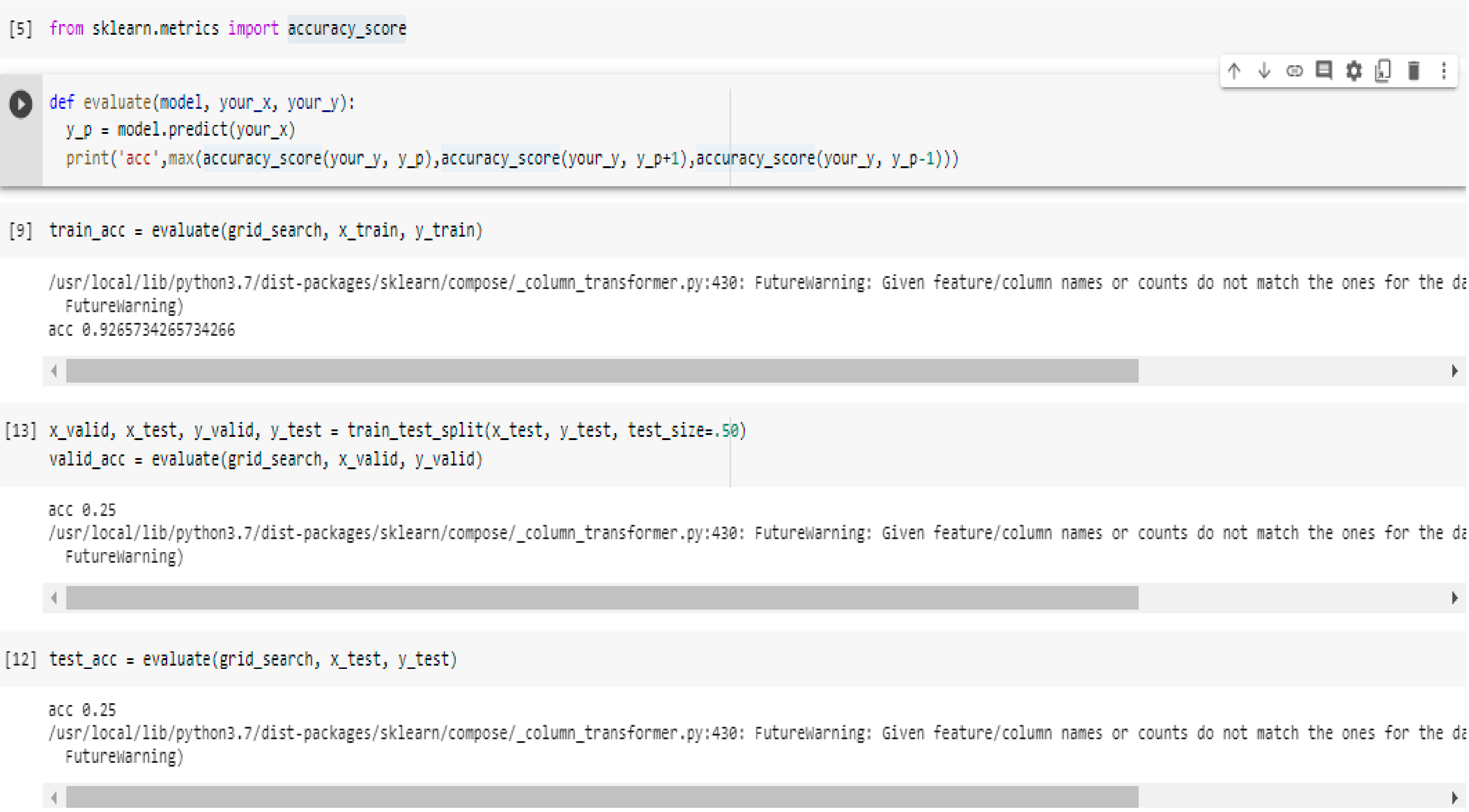
Overall, I extract the top music in 2019 from the basic dataset as a test dataset. I know this process could be a little bit weird but in order to show the relationship between music features and times, I think this one is better that we do not provide the data in 2019 and use the model to see whether we could find the trends. The next step is to preprocess the dataset as I mentioned above. The following step is to adjust and choose the hyperparameters for the model and apply grid search to generate the model and test its accuracy.

Experimental Results



```
Fitting 6 folds for each of 4 candidates, totalling 24 fits
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_split.py:667
  % (min_groups, self.n_splits)), UserWarning)
[Parallel(n_jobs=2)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done  24 out of  24 | elapsed:   18.3s finished
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:235:
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:268:
  y = column_or_1d(y, warn=True)
best score 0.25007196417320227
```

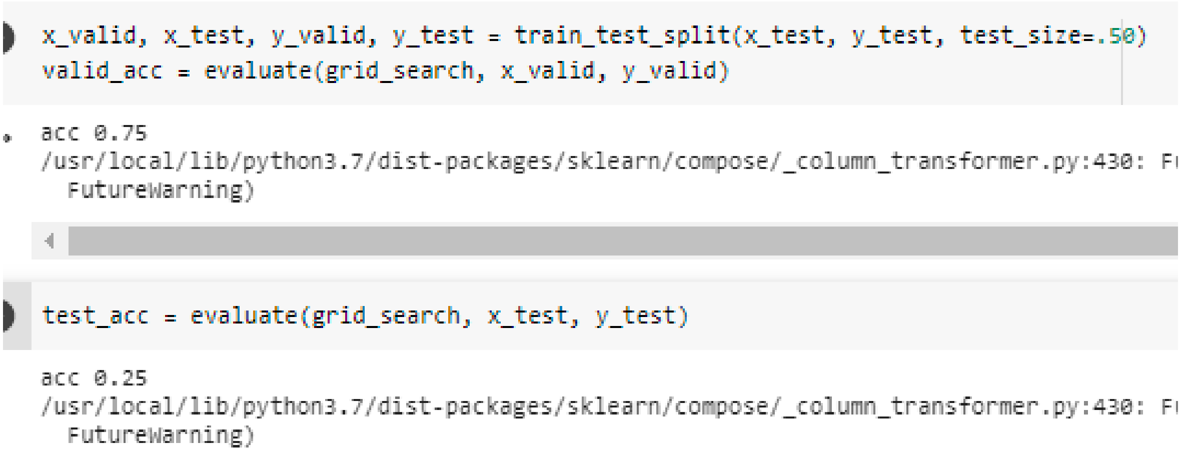
The result shows that our model does not fit the test set very much. Only around a quarter of a hundred percent. After several trials, the hyperparameters which can perform the best result are 'nrgy' and 'dnce' for numeric features and 'top genre' for the categorical feature. 'energy' represents the energy of a song and 'dnce' represents the danceability of the song. I also tried to ignore the categorical features. Nevertheless, the performance lowered after the operation which only contributes accuracy for 0.19098023663499566. This shows the categorical feature is still necessary.



```
[5]  from sklearn.metrics import accuracy_score

def evaluate(model, your_x, your_y):
  y_p = model.predict(your_x)
  print('acc',max(accuracy_score(your_y, y_p),accuracy_score(your_y, y_p+1),accuracy_score(your_y, y_p-1)))

[9]  train_acc = evaluate(grid_search, x_train, y_train)

/usr/local/lib/python3.7/dist-packages/sklearn/compose/_column_transformer.py:430: FutureWarning: Given feature/column names or counts do not match the ones for the da
  FutureWarning)
acc 0.9265734265734266

[13] x_valid, x_test, y_valid, y_test = train_test_split(x_test, y_test, test_size=.50)
     valid_acc = evaluate(grid_search, x_valid, y_valid)

acc 0.25
/usr/local/lib/python3.7/dist-packages/sklearn/compose/_column_transformer.py:430: FutureWarning: Given feature/column names or counts do not match the ones for the da
  FutureWarning)

[12] test_acc = evaluate(grid_search, x_test, y_test)

acc 0.25
/usr/local/lib/python3.7/dist-packages/sklearn/compose/_column_transformer.py:430: FutureWarning: Given feature/column names or counts do not match the ones for the da
  FutureWarning)
```

I try to figure out which part may lead to the inaccuracy. The accuracy of the training model is not bad which indicates the bias on this part is not obvious although, for this specific problem, no music song should be treated as bias from my perspective because every music work is unique in the world.

The next test for the validation of the testing set appears the obstacle. This tells us the model we select is not feasible which coincidence with the inference we made before. However, after multiple tests, I found the accuracy of the testing set is extremely unstable. Sometimes it could reach very high accuracy even a hundred percent) and sometimes it could be zero as the figures showed. So probably the main problem is that the testing set is too small that cannot validate our model. Although I tried to enlarge the testing set by adding top songs in 2018, the same problem still exists. So I guess we need to input more data to build sample data set with a far bigger size if we would like to fix the problem (Steven, 2021).

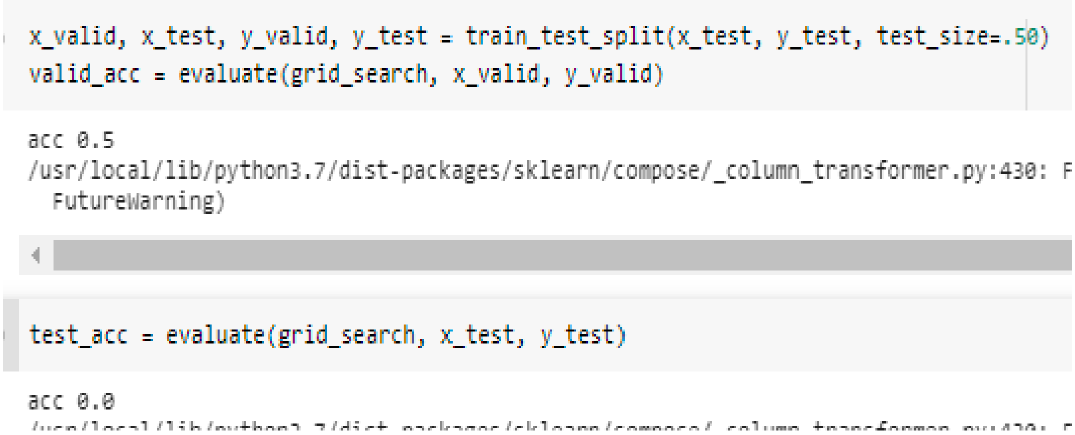


```
x_valid, x_test, y_valid, y_test = train_test_split(x_test, y_test, test_size=.50)
valid_acc = evaluate(grid_search, x_valid, y_valid)
```

```
acc 0.75
/usr/local/lib/python3.7/dist-packages/sklearn/compose/_column_transformer.py:430: F
  FutureWarning)
```

```
test_acc = evaluate(grid_search, x_test, y_test)
```

```
acc 0.25
/usr/local/lib/python3.7/dist-packages/sklearn/compose/_column_transformer.py:430: F
  FutureWarning)
```



```
x_valid, x_test, y_valid, y_test = train_test_split(x_test, y_test, test_size=.50)
valid_acc = evaluate(grid_search, x_valid, y_valid)
```

```
acc 0.5
/usr/local/lib/python3.7/dist-packages/sklearn/compose/_column_transformer.py:430: F
  FutureWarning)
```

```
test_acc = evaluate(grid_search, x_test, y_test)
```

```
acc 0.0
/usr/local/lib/python3.7/dist-packages/sklearn/compose/_column_transformer.py:430: F
```

Summary or Conclusion

In the previous result analysis, I mentioned that the energy of songs and danceability of songs are chosen as the hyperparameters. This result meets part of my expectation which means with the time flows, people focus more on the energy and danceability of songs. To be compared with the old days, people nowadays may regard music songs as part of party dance entertainment instead of a way to release their moods and express themselves. Another interesting result is that over 50% of the top songs are 'dance pop' in the 'top genre' categorical feature and more than 65% are pop songs.   Electronic pop music does not appear as much as I think even in the late 2010s. Although the model generating fails this time, we can jump to the conclusion that people in the 2010s prefer dance-pop songs more than any other type of songs else.

Besides, at first, I think the art could not be represented by data science. I changed my view through this experiment a little bit yet since the tests for the training set reach high accuracy. Perhaps if the dataset is large enough, we could generate a model to predict the era of the music in some day (or it has happened already?). The limitation to the size of the dataset is also a reason that I refuse to use a neural network which it cannot hold for a very small model in my opinion. Another reason is the prediction of the model should be linear regression as expected, I found that is unnecessary to use a neural network.

In contrast, some numeric features I have great confidence with do not show good performances in the relationship between the development of music and times. For example, the beats per minute and DB. One in my imagination should be more and more frequent and the other one should be louder and louder with time flows. For this part, I got an idea that in the early 2010s, there are still some songs in rock & roll that have frequent beats and loud sounds are very popular. And the data does show the bpm has its peaks in the early and late 2010s.

In brief, the model building is not ideal but shows some patterns that could be followed with the times. Music trends may be very changeable in one specific time with the ignites of some events.

Reference

Steven. Ding (2021, March 23) A2.ipynb Retrieved from https://github.com/CISC-372/Notebook