Aegis School of Data Science,

Mumbai

Project Report I

Spotify Sequential Skip Prediction

(Classification)

Submitted By :

Nihar Shah

Pravesh Raikwar

(DF-2009-CM)

Abstract

In this project, we experiment with real world dataset for classification. We explore few machine learning classification algorithms to fit the data. We were expected to gain experience using scikit ML library and how different algorithms works over a specific data. We have to explore the dataset using EDA processes then fit multiple models and tune the hyper parameters to get the maximum accuracy. Spotify is an online music streaming service with over 190 million active users interacting with a library of over 40 million tracks. A central challenge for Spotify is to recommend the right music to each user. A user skips a track is an important implicit feedback signal.

After performing required tasks, herein lies our final project report.

1.Introduction :

We have selected classification challenge that is hosted on www.aicrowd.com. In this challenge, we have to predict whether individual tracks encountered in a listening session will be skipped by a particular user. In order to do this, complete information about the first half of a user's listening session is provided, while the prediction is to be carried out on the second half. For this we have plenty amount of data from Spotify Technology.

About the scope of our problem we can understand it by understanding Spotify, Spotify is a digital music, podcast, and video streaming service that gives more than 180 million active users access to millions of songs. As we can see the problem is hosted by a music streaming company for its real world use, this problem is going to give us some real experience for solving real world problems.

2.Problem Definition

The task is to predict whether individual tracks encountered in a listening session will be skipped by a particular user. In order to do this, complete information about the first half of a user's listening session is provided, while the prediction is to be carried out on the second half. We have access to metadata, as well as acoustic descriptors, for all the tracks encountered in listening sessions. "Predict if users will skip or listen to the music they're streamed".

The output of a prediction is a binary variable for each track in the second half of the session indicating if it was skipped or not, with a 1 indicating that the track skipped, and a 0 indicating that the track was not skipped.

Official website : https://www.aicrowd.com/challenges/spotify-sequential-skip-prediction-challenge

3.Exploratory Data Analysis

3.1 Descriptive Analysis :

We have two csv files : 1.) log_mini.csv, 2.) tf_mini.csv

Log_mini.csv contain columns as follows :

#	Column	Non-Null Count	Dtype
0	session_id	167880 non-null	object
1	session_position	167880 non-null	int64
2	session_length	167880 non-null	int64
3	track_id_clean	167880 non-null	object
4	skip_1	167880 non-null	bool
5	skip_2	167880 non-null	bool
6	skip_3	167880 non-null	bool
7	not_skipped	167880 non-null	bool
8	context_switch	167880 non-null	int64
9	no_pause_before_play	167880 non-null	int64
10	short_pause_before_play	167880 non-null	int64
11	long_pause_before_play	167880 non-null	int64
12	hist_user_behavior_n_seekfwd	167880 non-null	int64
13	hist_user_behavior_n_seekback	167880 non-null	int64
14	hist_user_behavior_is_shuffle	167880 non-null	bool
15	hour_of_day	167880 non-null	int64
16	date	167880 non-null	object
17	premium	167880 non-null	bool
18	context_type	167880 non-null	object
19	hist_user_behavior_reason_start	167880 non-null	object
20	hist_user_behavior_reason_end	167880 non-null	object

tf_mini.csv contain columns as follows :

```
#   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   track_id                 50704 non-null  object
 1   duration                 50704 non-null  float64
 2   release_year             50704 non-null  int64
 3   us_popularity_estimate   50704 non-null  float64
 4   acousticness             50704 non-null  float64
 5   beat_strength            50704 non-null  float64
 6   bounciness               50704 non-null  float64
 7   danceability             50704 non-null  float64
 8   dyn_range_mean           50704 non-null  float64
 9   energy                   50704 non-null  float64
 10  flatness                 50704 non-null  float64
 11  instrumentalness         50704 non-null  float64
 12  key                      50704 non-null  int64
 13  liveness                 50704 non-null  float64
 14  loudness                 50704 non-null  float64
 15  mechanism                50704 non-null  float64
 16  mode                     50704 non-null  object
 17  organism                 50704 non-null  float64
 18  speechiness              50704 non-null  float64
 19  tempo                    50704 non-null  float64
 20  time_signature           50704 non-null  int64
 21  valence                  50704 non-null  float64
 22  acoustic_vector_0        50704 non-null  float64
 23  acoustic_vector_1        50704 non-null  float64
 24  acoustic_vector_2        50704 non-null  float64
 25  acoustic_vector_3        50704 non-null  float64
 26  acoustic_vector_4        50704 non-null  float64
 27  acoustic_vector_5        50704 non-null  float64
 28  acoustic_vector_6        50704 non-null  float64
 29  acoustic_vector_7        50704 non-null  float64
```

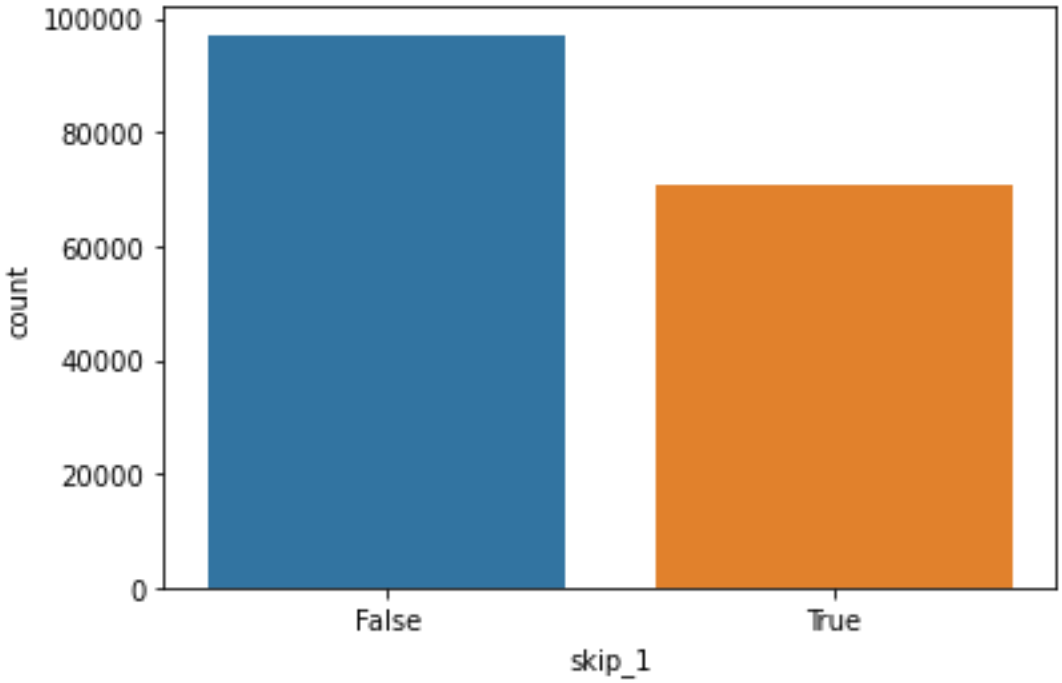
We have 167880 data points consisting 51 features after merging these two datafile on track_id.

Skewness in data:

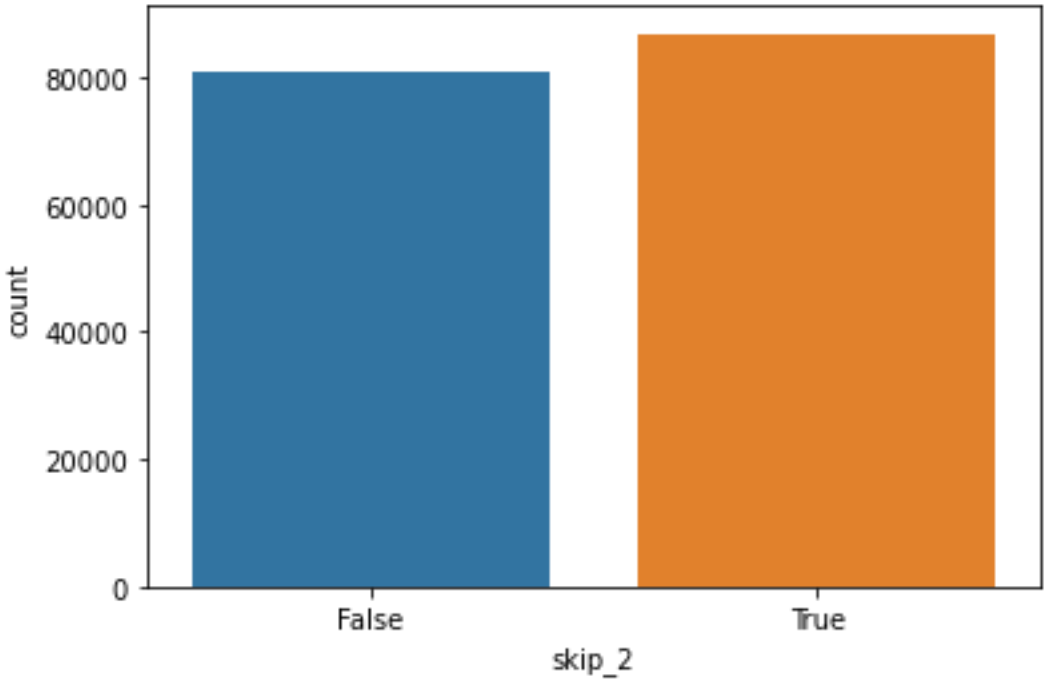
```
session_position                  0.250537
session_length                   -1.093022
skip_1                            0.317931
skip_2                           -0.068757
skip_3                           -0.621523
not_skipped                       0.709275
context_switch                    4.635773
no_pause_before_play             -1.267181
short_pause_before_play           1.997894
long_pause_before_play            1.730598
hist_user_behavior_n_seekfwd     53.436156
hist_user_behavior_n_seekback   122.850478
hist_user_behavior_is_shuffle     0.828058
hour_of_day                      -0.699712
premium                          -1.583901
duration                          2.220571
release_year                     -3.242944
us_popularity_estimate           -6.242498
acousticness                      1.316370
beat_strength                    -0.120968
bounciness                       -0.325705
danceability                     -0.544787
dyn_range_mean                    0.424233
energy                           -0.378494
flatness                         -2.777386
instrumentalness                  5.148110
key                               0.041289
liveness                          2.236309
loudness                         -2.412514
mechanism                        -0.506825
organism                          0.736462
speechiness                       1.889123
tempo                             0.172380
time_signature                   -5.041217
valence                           0.255647
acoustic_vector_0                 0.723081
acoustic_vector_1                -1.902545
acoustic_vector_2                -1.566726
acoustic_vector_3                -0.137237
acoustic_vector_4                 1.284050
acoustic_vector_5                -0.777207
acoustic_vector_6                 1.707008
acoustic_vector_7                -1.106678
```

Some of the columns have skewness in it.Some are positively skewed columns and some are negatively.
Negatively Skewed columns [session_length, no_pause_before_play, premium, release_year, us_popularity_estimate, loudness, acoustic_vector_1,2,7]

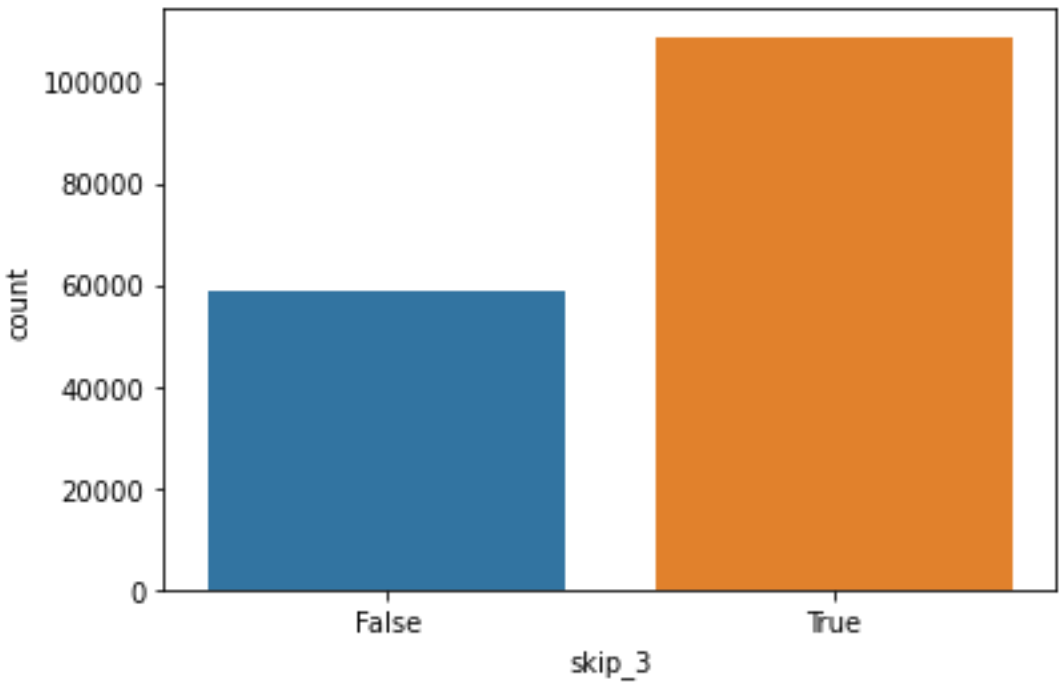
3.2 Data Visualizations:



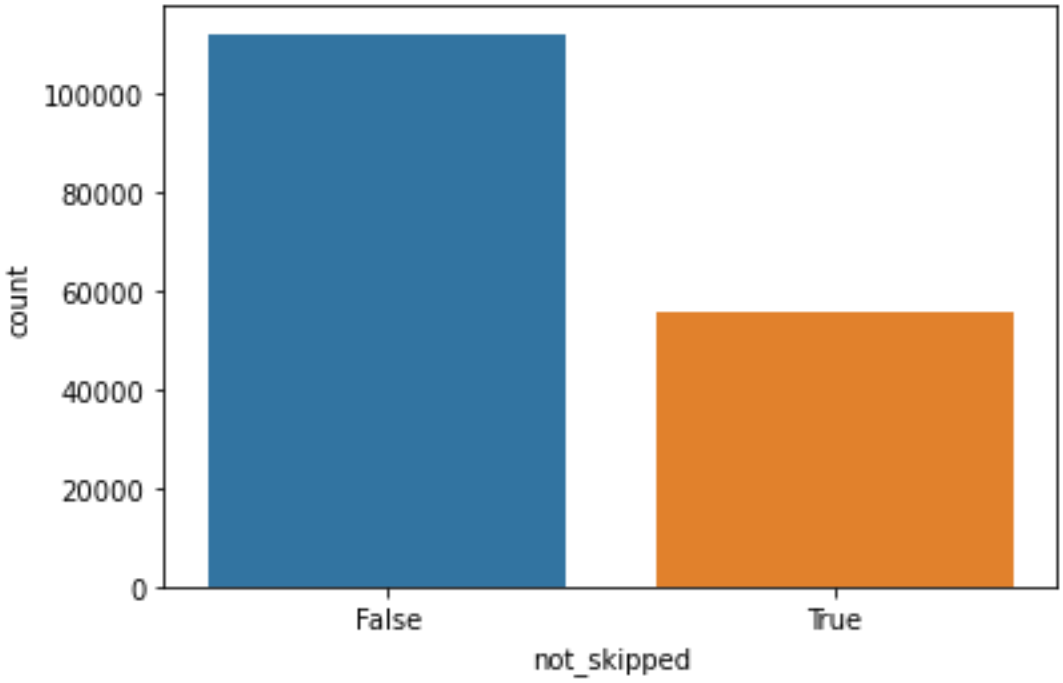
We can see here that skip_1 (people skip the song at very starting of the song) is less skiped, that mean skipping of song at very starting of the song is less.



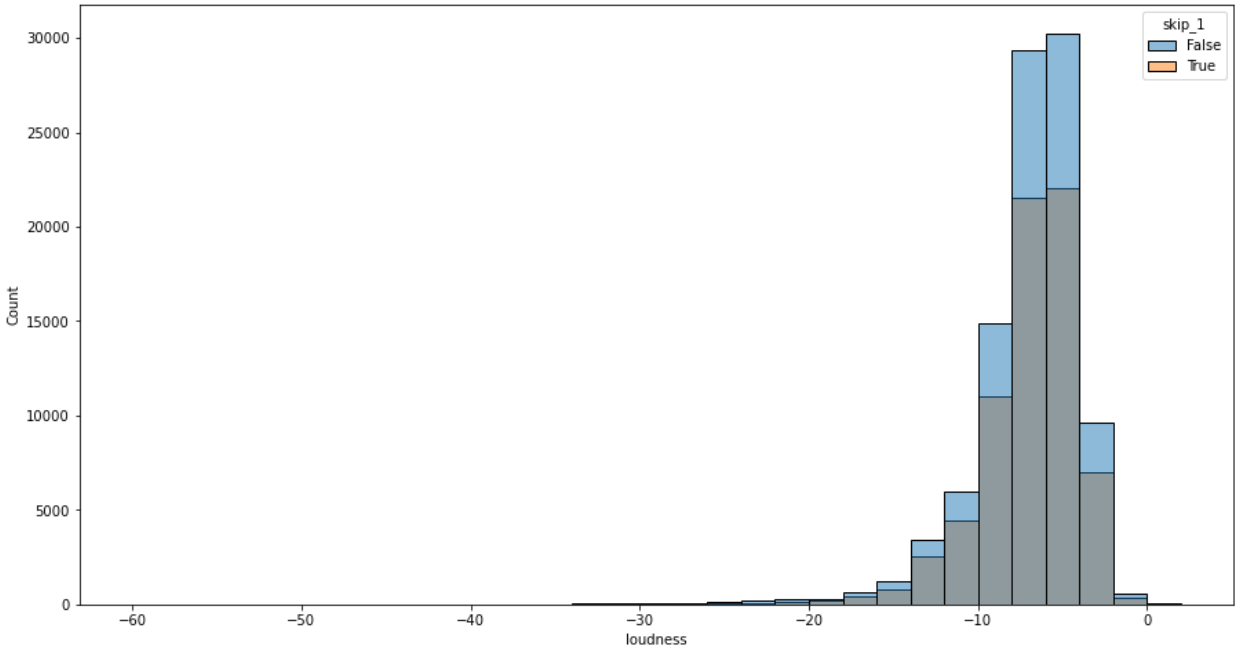
User's skip_2 is almost same, 50% chances that user will skip the song.



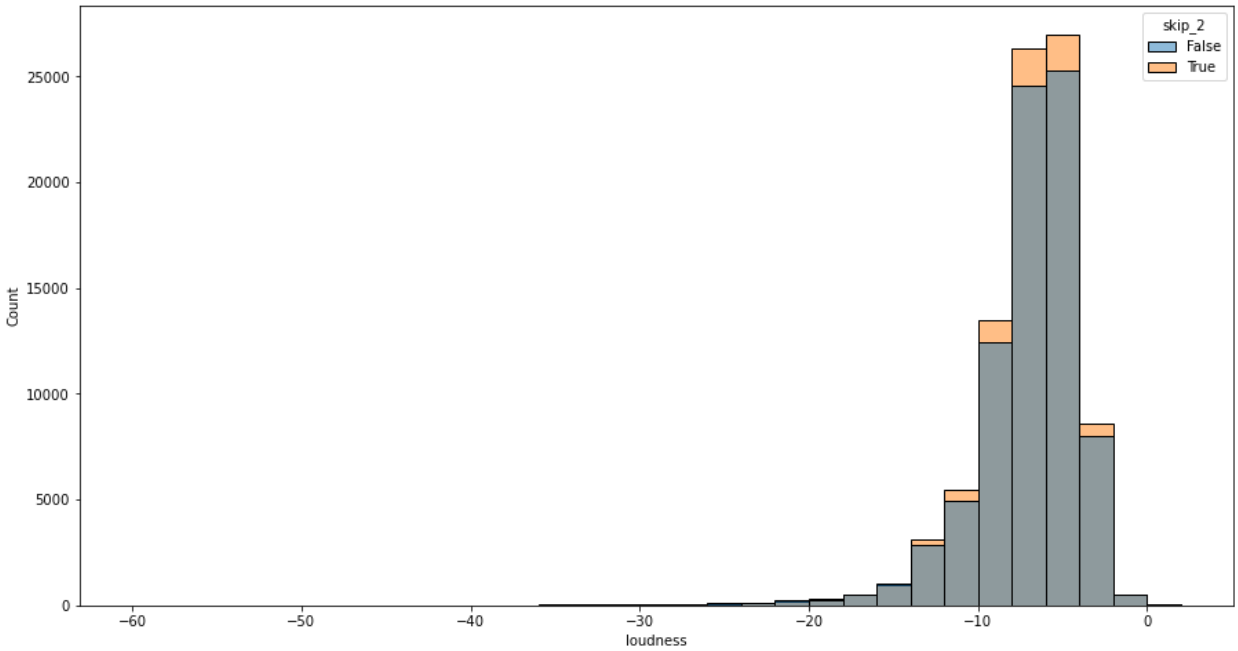
Here, most of the song skiped at the very end of song.

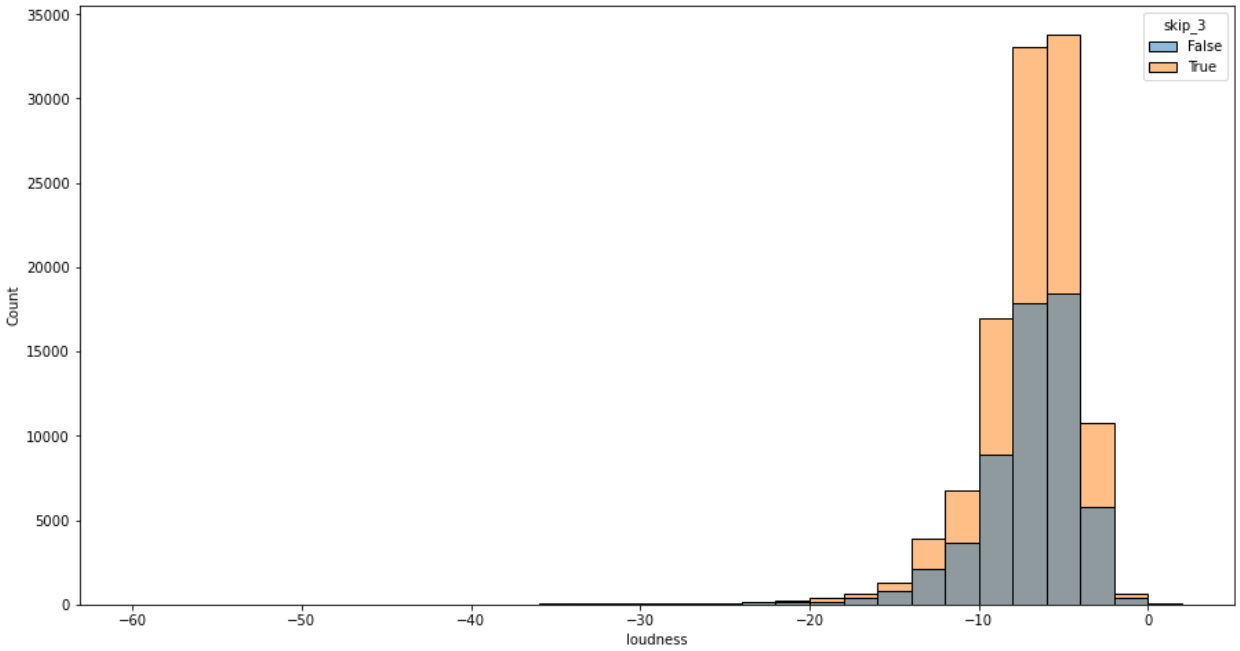


Here, most of the song didn’t get skiped.

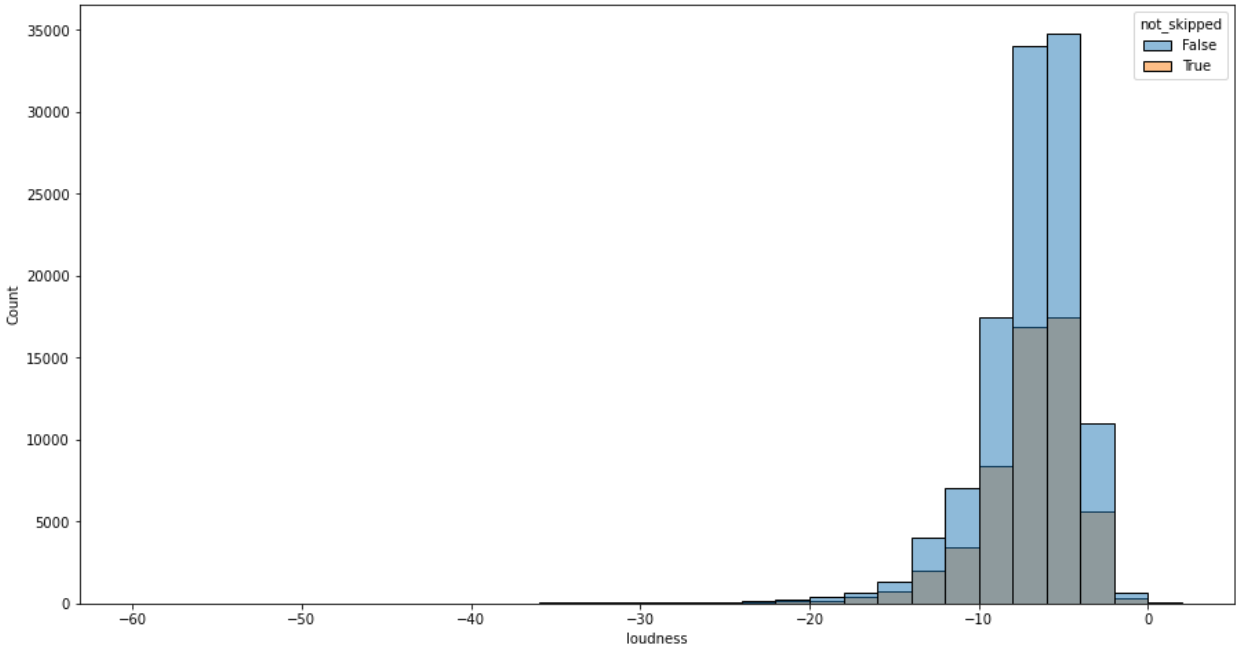


if loudness is in range of -4 to -8, most of the people dont skip the song in 1st phase.

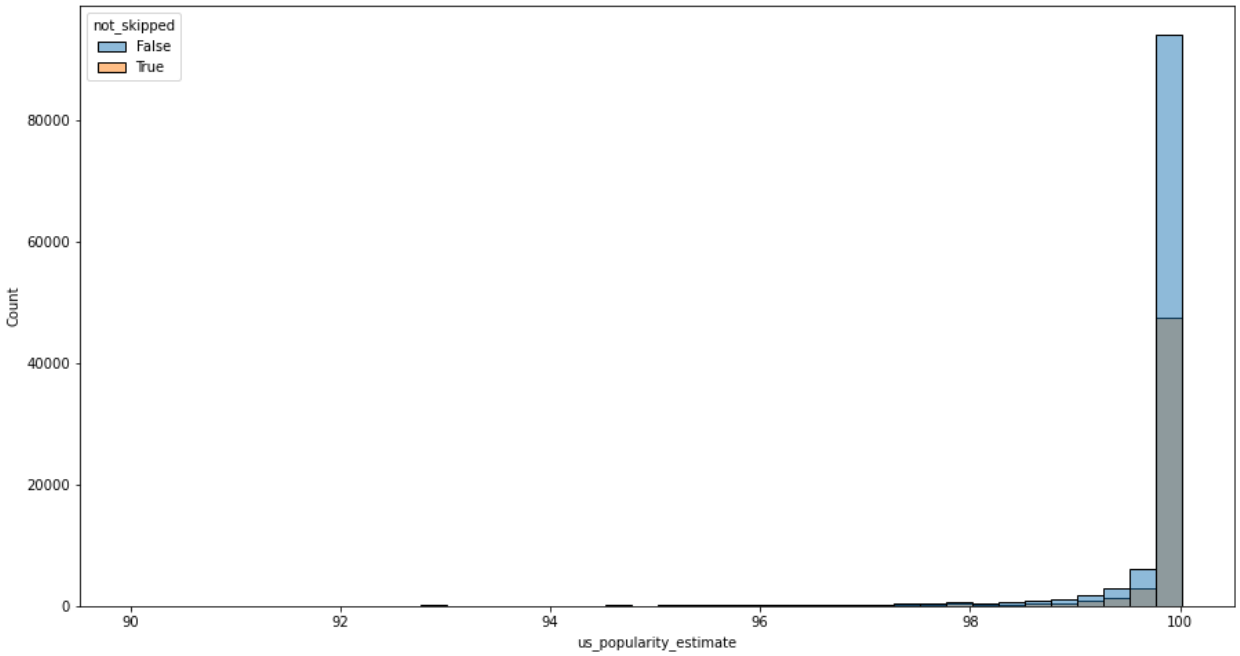




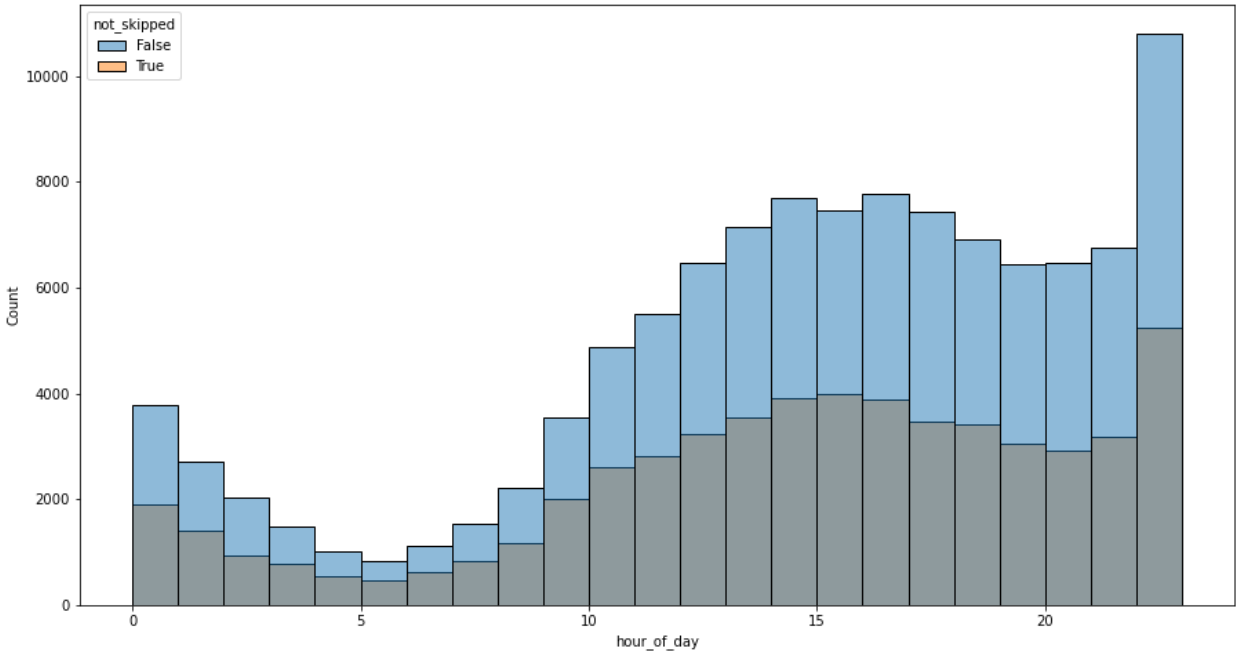
for any loadness value song is getting skipped in 2nd or 3rd phase



this proves that if value of loudness is less than zero, most of the songs gets skipped.

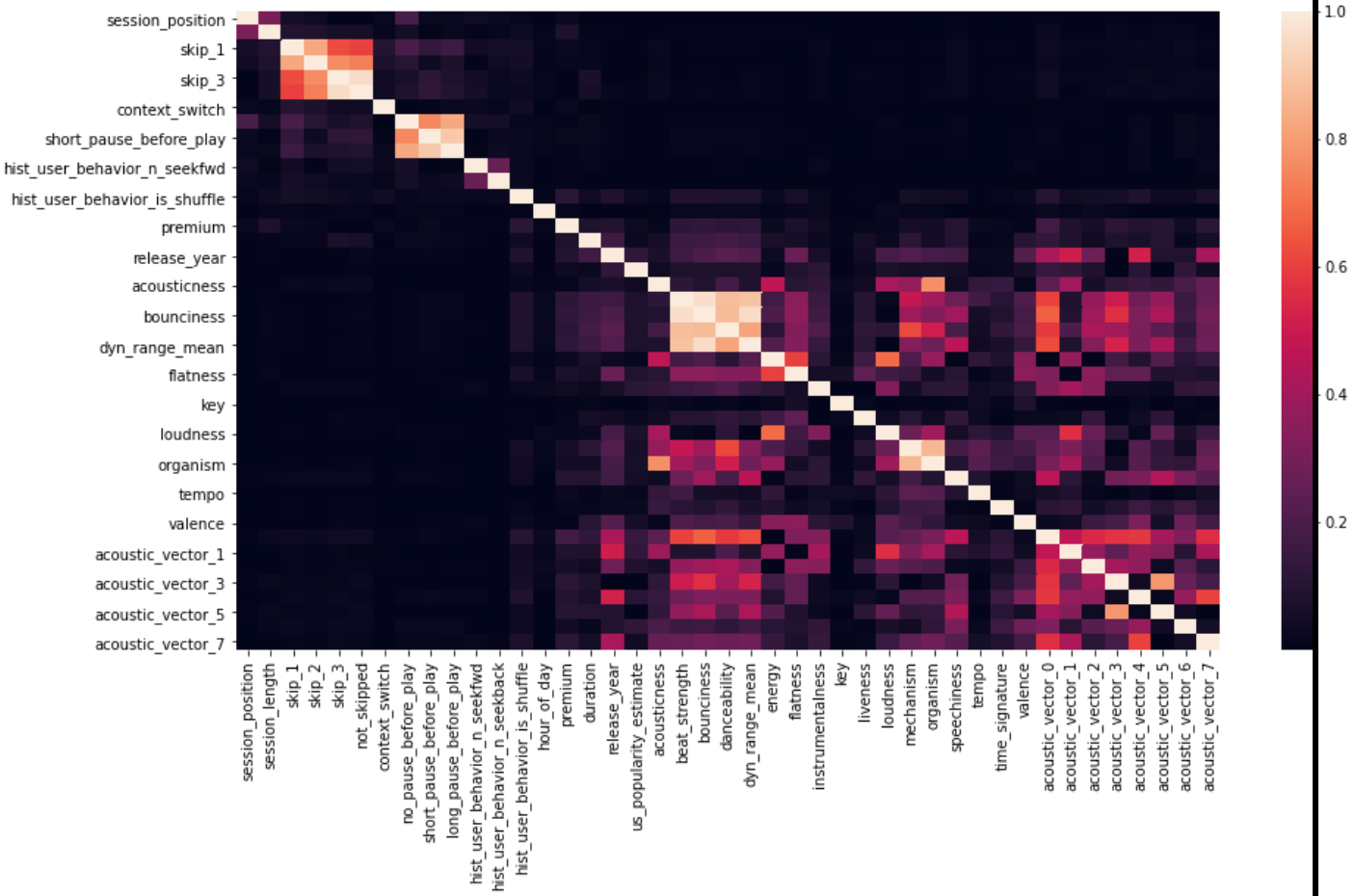


this proves that if value of loudness is less than zero, most of the songs gets skipped.



The ratio of song getting skipped is around 50% for each hour of day. around 12 midnight, people listen the song most.

Heatmap :



There is no clear interpretation of feature that are linearly corelated. But we can infer via lightly shaded area, that there is some collinearity in the dataset.

3.3 Feature Selection :

Principal Component Analysis:

PCA is an unsupervised statistical technique that is used to reduce the dimensions of the dataset. ML models with many input variables or higher dimensionality tend to fail when operating on a higher input dataset. PCA helps in identifying relationships among different variables & then coupling them. After applying pca we got principal Componentas as:

```
In [43]: np.cumsum(pca.explained_variance_ratio_)

Out[43]: array([0.76835394, 0.95663865, 0.97628108, 0.98389601, 0.99056774,
                0.99343847, 0.99555385, 0.99757501, 0.99888193, 0.99904781,
                0.99917939, 0.99927543, 0.99936668, 0.99944784, 0.99951858,
                0.99957132, 0.99961666, 0.99965589, 0.99969076, 0.99972403,
                0.99975533, 0.999783  , 0.99980811, 0.99983224, 0.99985503,
                0.99987491, 0.99989279, 0.99990992, 0.99992167, 0.99993102,
                0.99994013, 0.99994778, 0.9999554 , 0.99996182, 0.99996737,
                0.99997235, 0.99997683, 0.99998126, 0.99998452, 0.99998713,
                0.99998958, 0.99999166, 0.99999353, 0.99999505, 0.99999614,
                0.99999704, 0.99999781, 0.99999832, 0.99999872, 0.99999904,
                0.9999993 , 0.99999948, 0.99999961, 0.99999971, 0.99999982,
                0.99999991, 0.99999996, 0.99999999, 1.        , 1.        ,
                1.        , 1.        , 1.        ])
```

Here, we selected 4 PCs that gives us 98.3% percent variability of total data. And dimensionality is reduced by many folds as we are using only 4 feature but our original data has more than 50 features.

4. Evaluate Algorithms

4. 1 **Logistic Regression :**

Models best parameters are as follows:
{'multi_class': 'ovr', 'solver': 'liblinear'}

Score :
lr_model.score(X_val, y_val)
LR with PCA data = 0.7186207352989841
LR with X and Y = 0.8049571461663192

4.2 **K-Nearest Neighbor :**

Models best prametes are as follows :
{'algorithm': 'kd_tree', 'n_neighbors': 37, 'p': 1, 'weights': 'distance'}

Scores :
KNN with PCA Data = 0.8054866143816805
KNN with X and Y = 0.6646811608590621

4.3 Random Forest :

Models best parameters :
{criterion='entropy', max_depth=12, n_estimators=1000, n_jobs=-1,
oob_score=True}

Scores:
RF with PCA Data --> 0.9274258398856325
RF with X and y ---------> 0.9304130926852514

4.4 Support Vector Machine :

Model paramters :
{ decision_function_shape='ovo', kernel='sigmoid'}

Scores :
SVM with PCA data ---->0.6389688606505841
SVM with X and y ------->0.4212912406102121

In case of SVM, when we are having high dimentionality in the data the model is not able to perform well. After applying PCA when we reduced the number of features(PCs) to 4 the model is starting to perform well. But it is still not able to find the pattern in the data that well. The best val score it can give is around 64%.