Midterm Project——

Music Recommendation

Author: Guangyan Yu

Date: 12/9/2018

Abstract

With increasing development of digital music database, automatic music recommendation has become an increasingly relevant problem in recent years. Most recommender systems rely on collaborative filtering, which based on users' interests without considering the content information of music. I try to combine both information of users and music, so that for new songs, I can determine the probability of recommendation based on its musical features like genres. In this paper, the problem I focused on is to predict whether a user will listen to a song repeatedly. The EDA part shows the overview of the whole data and the process I choosing demographic and musical features. In the model part, I fit logistic regression and multilevel logistic regression with the variables I chose by EDA and get similar result. Both of them get 0.65 AUC score.

1. Introduction

For music application company, apparently better recommendation system can attract more users. For example, "NeteaseMusic" has a high-quality music recommendation for users everyday, and a lot of people use this music app because of its accurate recommendation. To recommend personally, there are three main types of recommendation models used by companies: 1. Collaborative Filtering models (i.e. the ones that Last.fm originally used), which analyze both your behavior and others' behaviors. 2. Natural Language Processing (NLP) models, which analyze text. 3. Audio models, which analyze the raw audio tracks themselves. In this paper, I just do the simple linear regression to fit the data.

2. Method

2.1 Data source

The dataset is from KKBOX, Asia's leading music streaming service, holding the world's most comprehensive Asia-Pop music library with over 30 million tracks.

song-user.csv

- msno: user id
- song_id: song id
- source_system_tab: the name of the tab where the event was triggered. System tabs are used to categorize KKBOX mobile apps functions. For example, tab "my library" contains functions to manipulate the local storage, and tab "search" contains functions relating to search.
- source_screen_name: name of the layout a user sees.

- source_type: an entry point a user first plays music on mobile apps. An entry point could be album, online-playlist, song .. etc.
- target: this is the target variable. target=1 means there are recurring listening event(s) triggered within a month after the user's very first observable listening event, target=0 otherwise.

songs.csv
- song_id
- song_length: in ms
- genre_ids: genre category. Some songs have multiple genres and they are separated by |
- artist_name
- composer
- lyricist
- language
- members.csv
- user information.

msno
- city
- bd: age. Note: this column has outlier values, please use your judgement.
- gender
- registered_via: registration method
- registration_init_time: format %Y%m%d
- expiration_date: format %Y%m%d

2.1.1 Overview

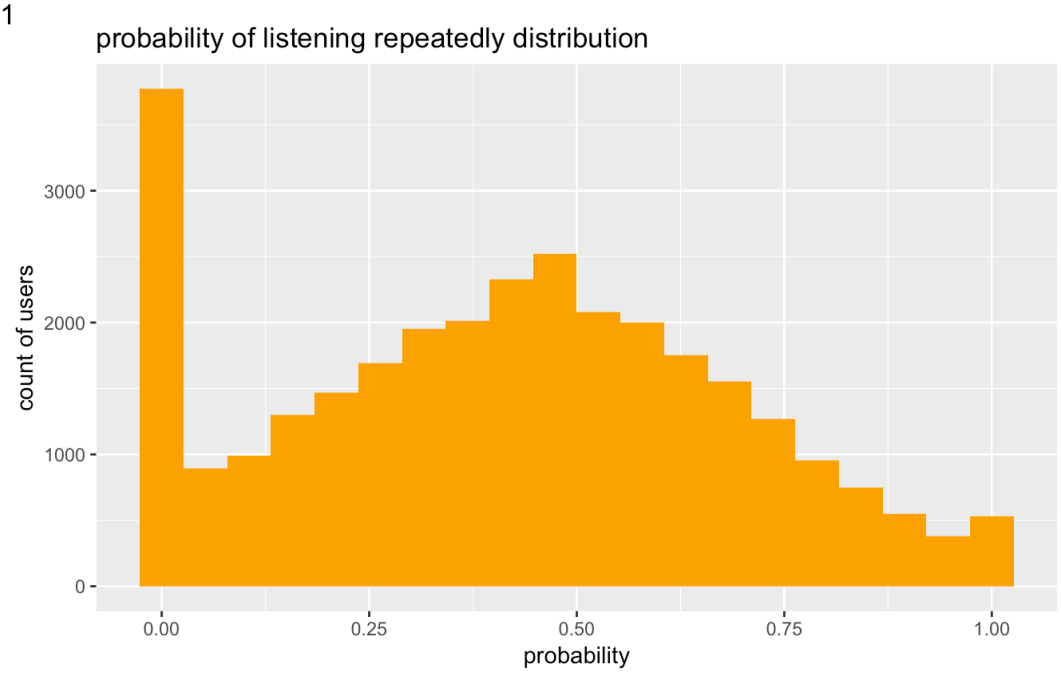
- "target" in song-user data is my response.
- There are 27179 users in the song-user dataset(around 700,0000 observations), everyone has different number of history data. I separate the data in train set and test set by 3:1.
- There are 20468 overlapping users in train and test set, which means that 85% users in test data set are old users.
- There are 2481398 user-song pairs of no-repeated; 2896020 user-song pairs of repeated, which means that the data of two groups(repeat and non repeat) has no bias.
- There are 294637 unique songs in train dataset, and 47 of them do not have data in "song.csv", so I removed them from test set.
- There are 147 unique genres in the songs with single genre, contains unknown genre, and 385 mixed genres. Only 83 observations in test set has new genre_ids, I removed them.

2.1.2 EDA

1. msno(user_id)

I assume that every user has his or her own habit, which means some users prefer to listen to songs repeatedly while others not. So I see the distribution of listening repeatedly probability to find if there is difference between users.

fig.1

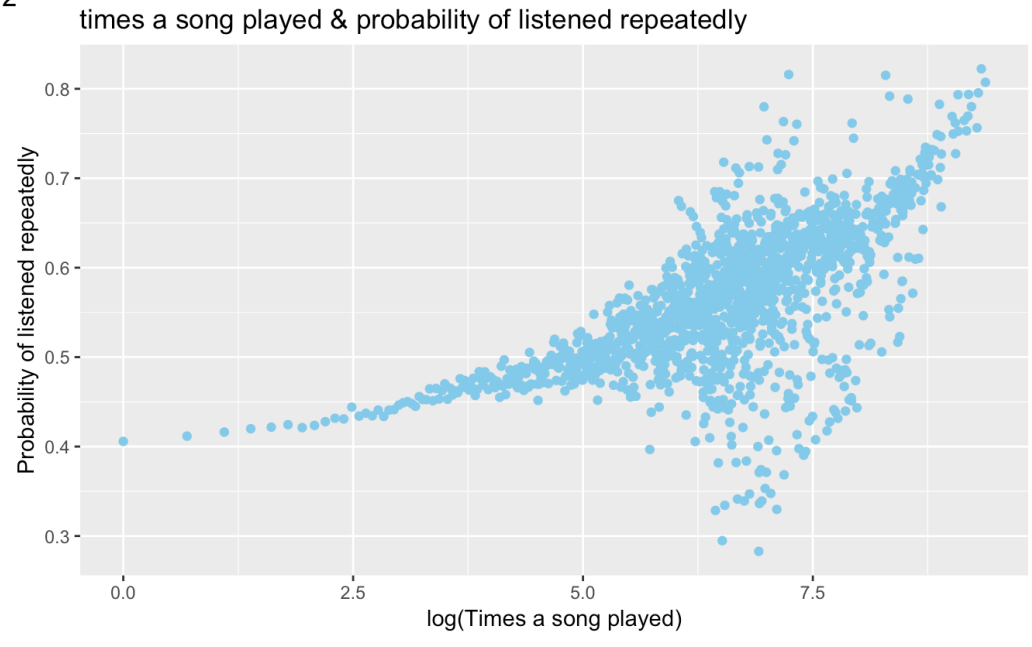


- *The distribution plot shows that users have individual habit, a great number of users will not listen to songs repeatedly, but someone tend to listen to songs repeatedly.*

- *msno(user_id) can be a predictor for target. For multilevel model, group by msno(user_id) is sensible.*

2. Times a song played

fig.2



- *The plot shows that when songs are played more times, the corresponding probability of the song repeated increases. But for the songs played more than exp(5) times, though the positive linear relationship is still exist, there is more noise than songs played less times.*

- *Times a song played can be a predictor for target.*

3. ui information

Table 1: number of observations and the probability of songs listened repeatedly for each ui tab

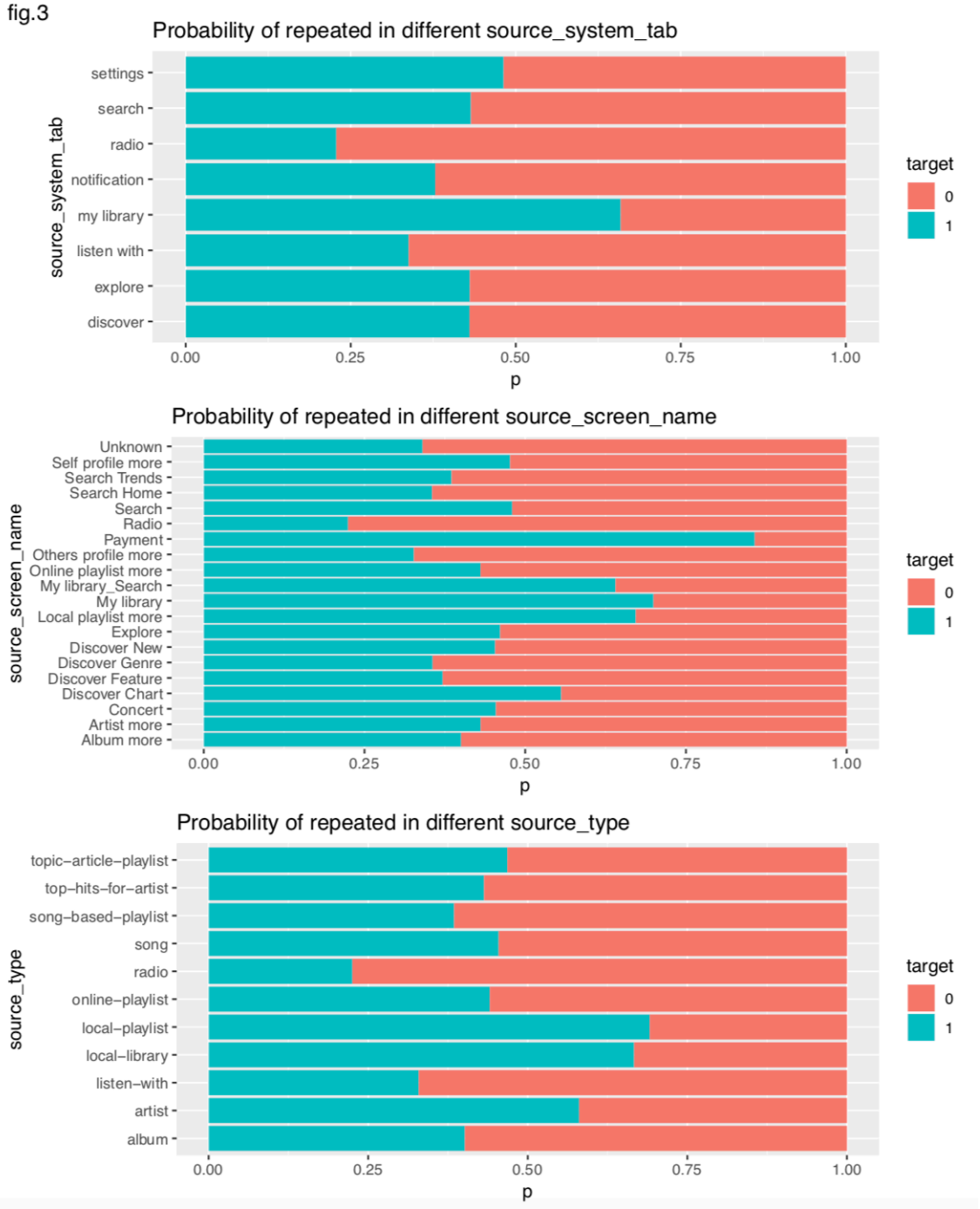
source_system_tab	count	p
my library	1837602	0.6590455
settings	328	0.4816446
search	177334	0.4320846
explore	44069	0.4304706
discover	572181	0.4299756
notification	1541	0.3780667
listen with	47667	0.3378362
radio	70421	0.2282069

- *Most songs are played from "my library" and "discover"*

- *songs played from "my library"" are most likely to be listened repeatedly, and different tabs have significant differences.*

- *source_system_tab is correlated to whether the song would be listened repeatedly.*

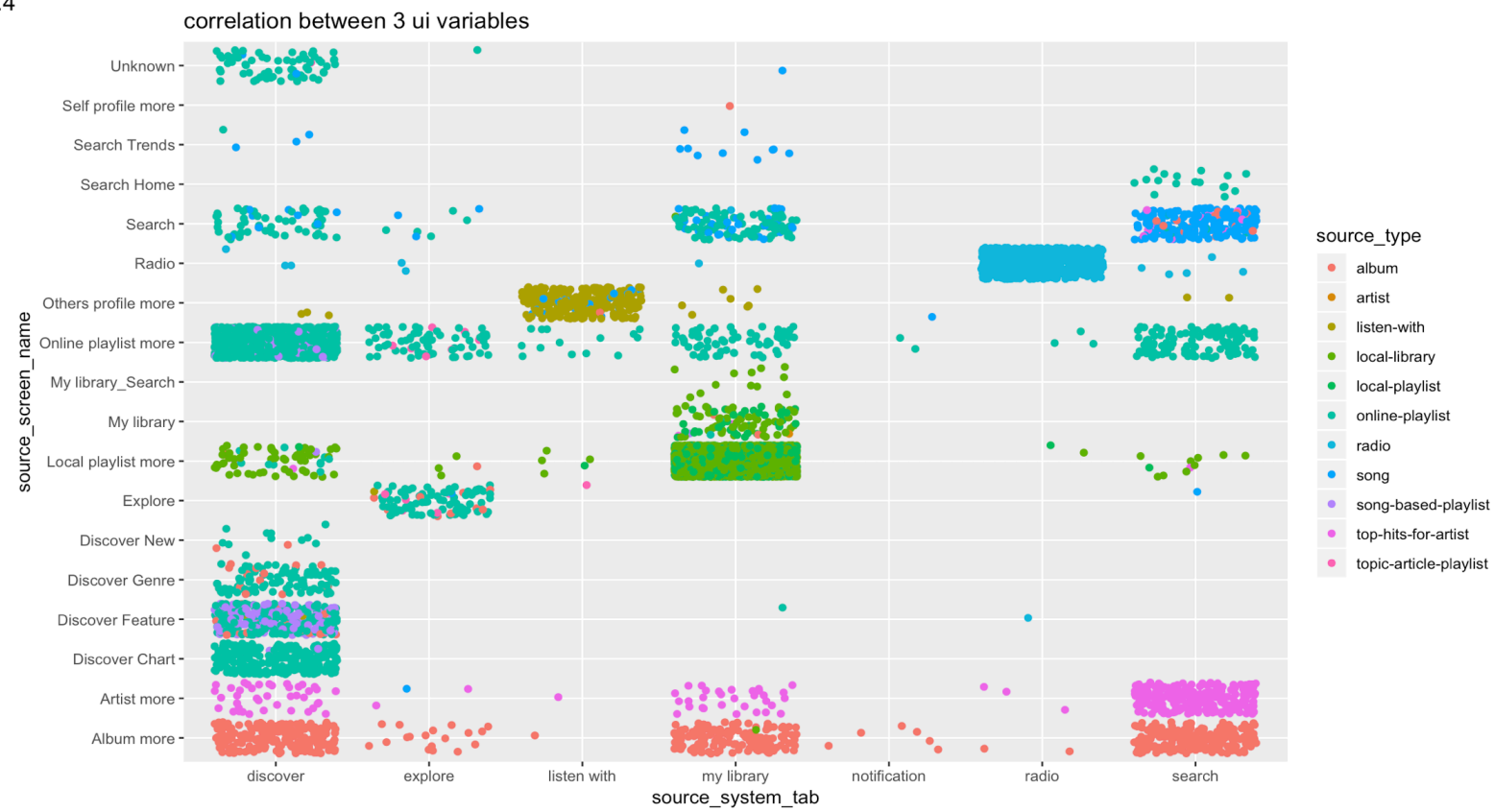
fig.3



The bar plot shows that also screen_name and source_type have similar appearance with tab and are correlated to the probability of songs listened repeatedly.

In such case, I want to see if these three UI attributes are correlated.

fig.4

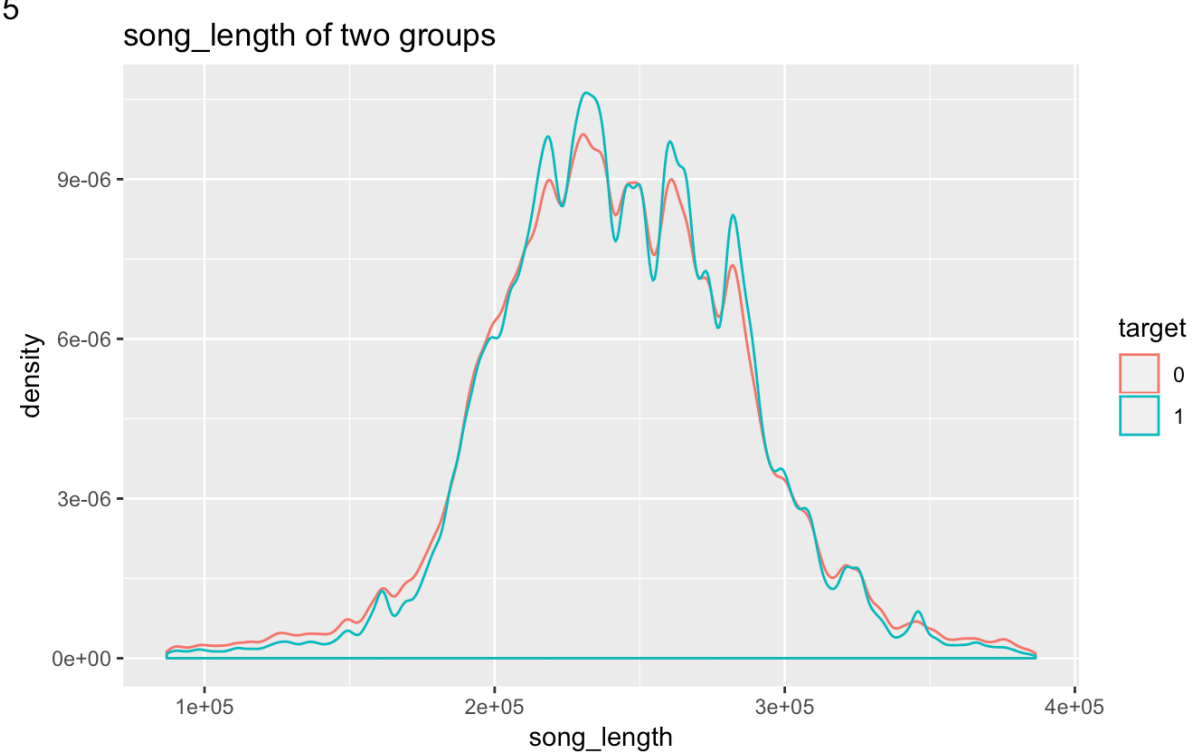


- *There are some clusters in the plot, and most points in the clusters have the same color, which means that there is some correlation in these three variables.*

- *So in the model, I choose "source_system_tab" from these three as a predictor.*

4. song length

fig.5



- *The density plot shows there is a little difference in song_length between songs repeated and no-repeated. For songs having length in (200000,300000), they are more likely to be repeated. For songs having length in (100000,200000), they are more likely to listened non-repeatedly.*

- *So when fitting the model, we first choose it as a predictor and see whether it is significant.*

5. genre

```
        Pearson's Chi-squared test

data:  g_new
X-squared = 41336, df = 516, p-value < 2.2e-16
```

Through the chisquare test for "genre_ids" and "target", we can see that p-value<0.05, which means the two variables are correlated.

Table 2: Probability of songs listened repeatedly with genres

genre_ids	p
458	0.5842614
465	0.5488282
1609	0.5478873
444	0.5471475
921	0.5328194
359	0.5071398
1259	0.5039161
2022	0.4501991

Then we choose the genres occurred more than 50000 times(data with more samples are more believable). From the table, we can see there is difference in probability of listening repeatedly between different genres. So "genre_ids" can be chosen as a predictor for target.

6. Age

fig.6

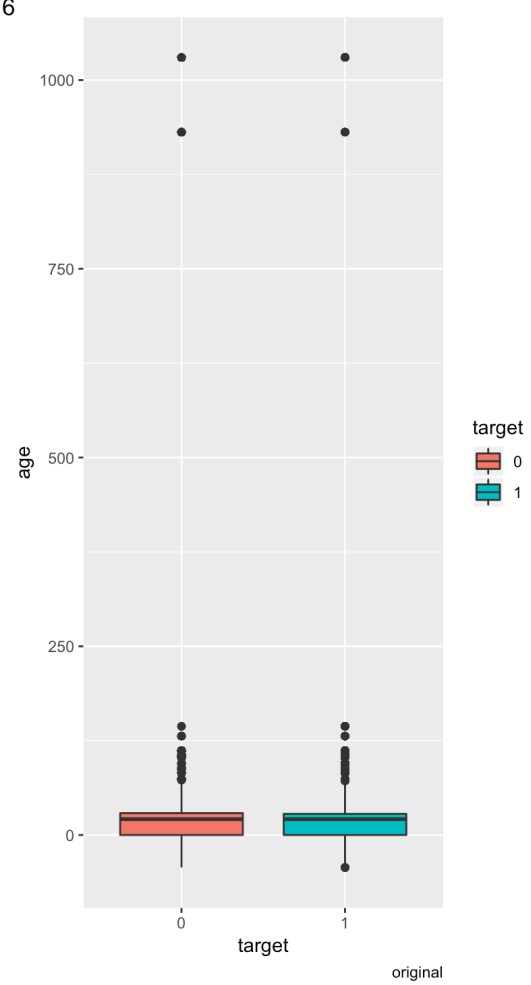
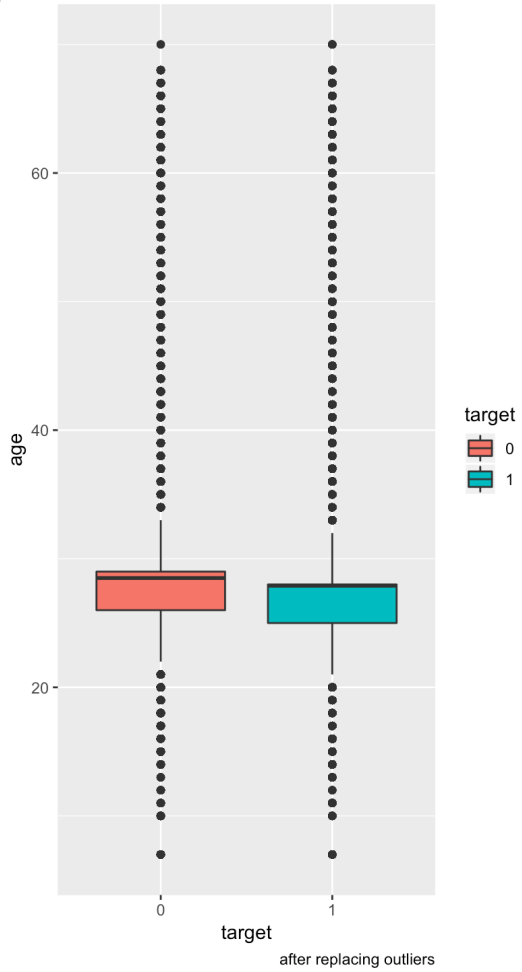


fig.7



- *The left boxplot shows that there are outliers in age data, we use average age of each group to replace the outliers.*
- *After replacing outlier ages with average ages of two groups, the boxplot shows that older users are less likely to listen to a song repeatedly.*
- *So we could choose age as an predictor.*

7. City

fig.8

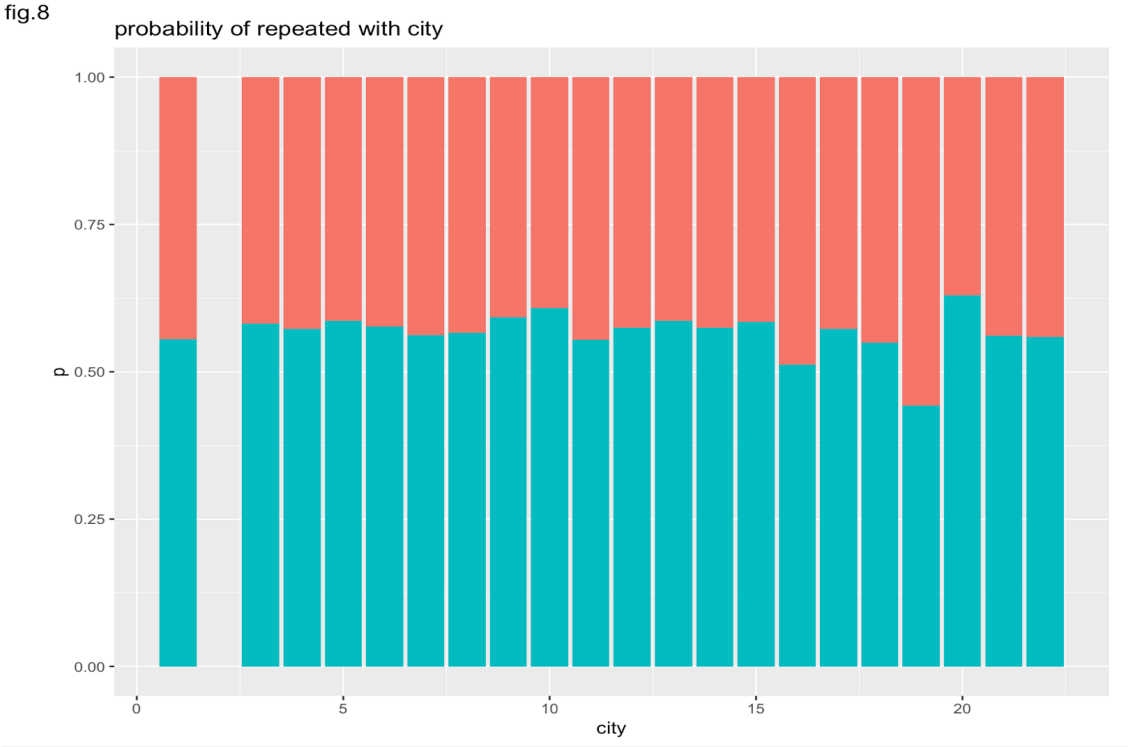


Table 3: City with probability of songs repeated

city	count	p
20	1667	0.6297696
10	16179	0.6077989
9	22982	0.5925028
5	177941	0.5865536
13	271268	0.5864113
15	112520	0.5846197
3	15260	0.5820207
6	61909	0.5765466
14	53687	0.5748873
12	33443	0.5744345
17	10025	0.5725954
4	123293	0.5725291
8	18680	0.5657349
7	6361	0.5623729
21	15498	0.5615421
22	99282	0.5594581
1	535490	0.5555106
11	15529	0.5544289
18	18845	0.5494329
16	1596	0.5120308
19	1066	0.4423237

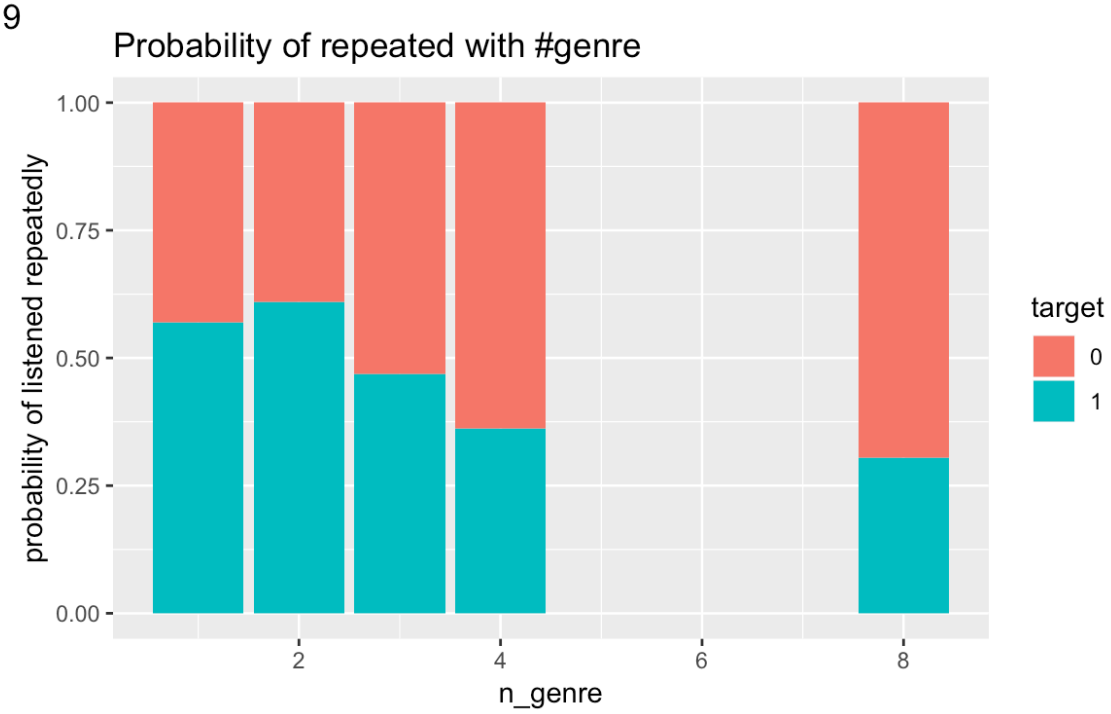
- *From the bar plot, city "16" "19" and "20" have extreme repeated probability compared with other cities. But from the table we notice that these cities only have 1596, 1066, 1667 observations, which is small compared to more than 200,0000 observations as total. So we can say that data of these three cities are biased. Except for the three cities, other cities do not have significant difference in probability of songs repeated.*
- *So I don't choose city as a predictor for target.*

8. number of genres a song have

Table 4: size of data with same number of genres

n_genre	count
1	2686328
2	134165
3	3274
4	1724
5	15
6	15
7	8
8	194

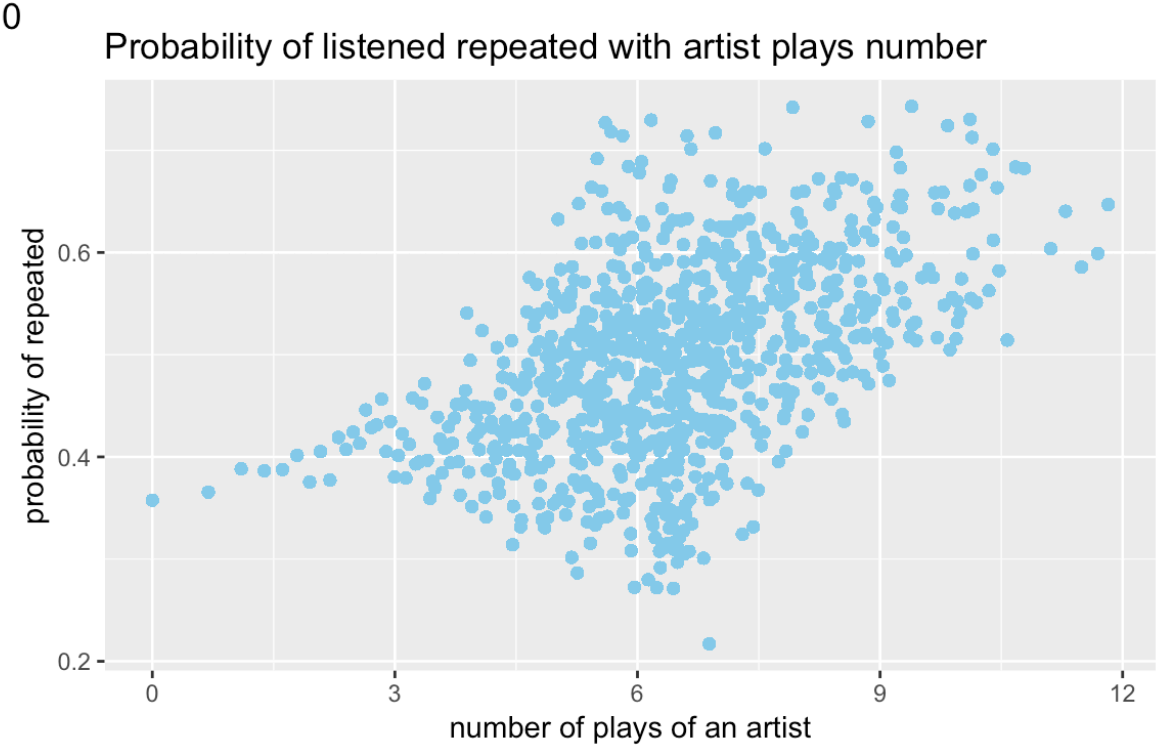
fig.9



- *When n_genre=5,6,7, the result is not believable because of small sample size, so we remove them.*
- *The plot shows that the number of genres is likely to have negative relationship with the probability of song repeated.*

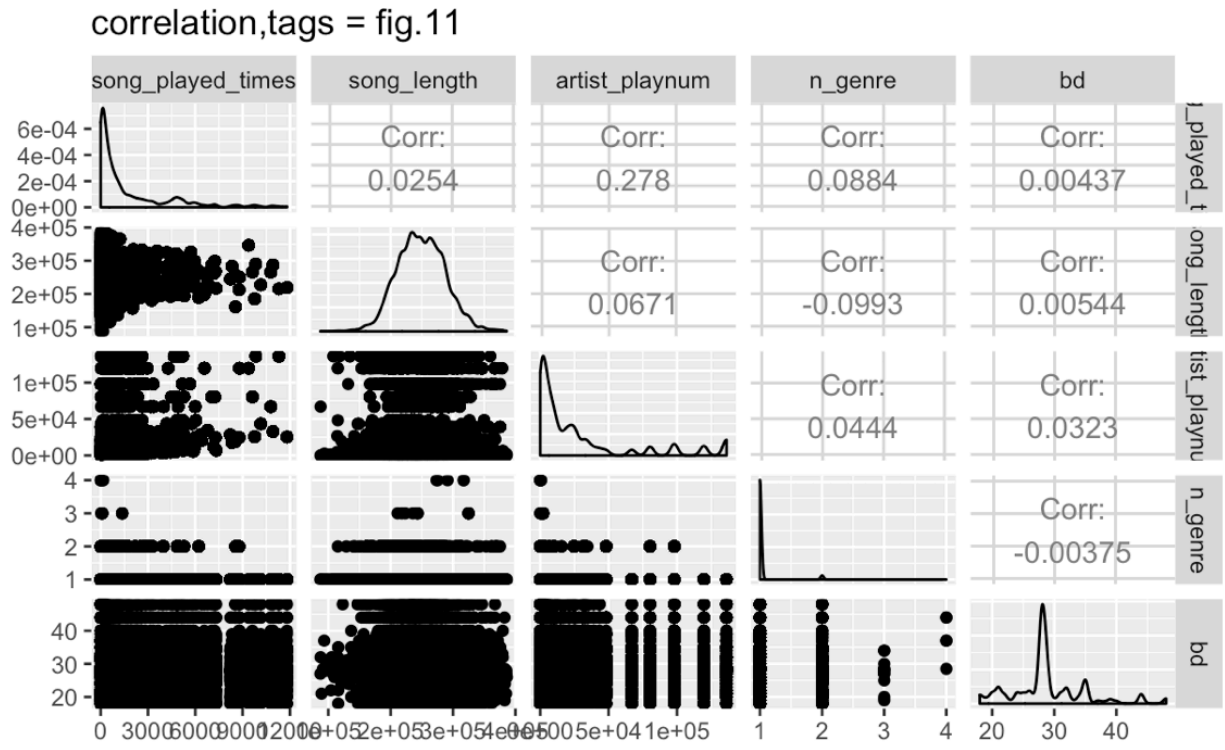
9. number of songs of an artist

fig.10



- *The plot shows that for songs whose artist has been played more times, the corresponding probability of the song repeated is bigger.*
- *Times a song played can be a predictor for target.*

10. Correlation between numeric variables



The correlation plot shows that the five numeric variables are not strong correlated, so I can choose them as my predictors.

2.2 Model Used

2.2.1 Model1(Logistic regression)

I first choose “msno", “source_system_tab”, “genre_ids", “song_played_times", “song_length", “artist_playnum", “bd”(age), "n_genre" as predictors.

Because the sample size is too big for fitting a linear regression, I sampled songs history data of 100 users, including 22116 observations.

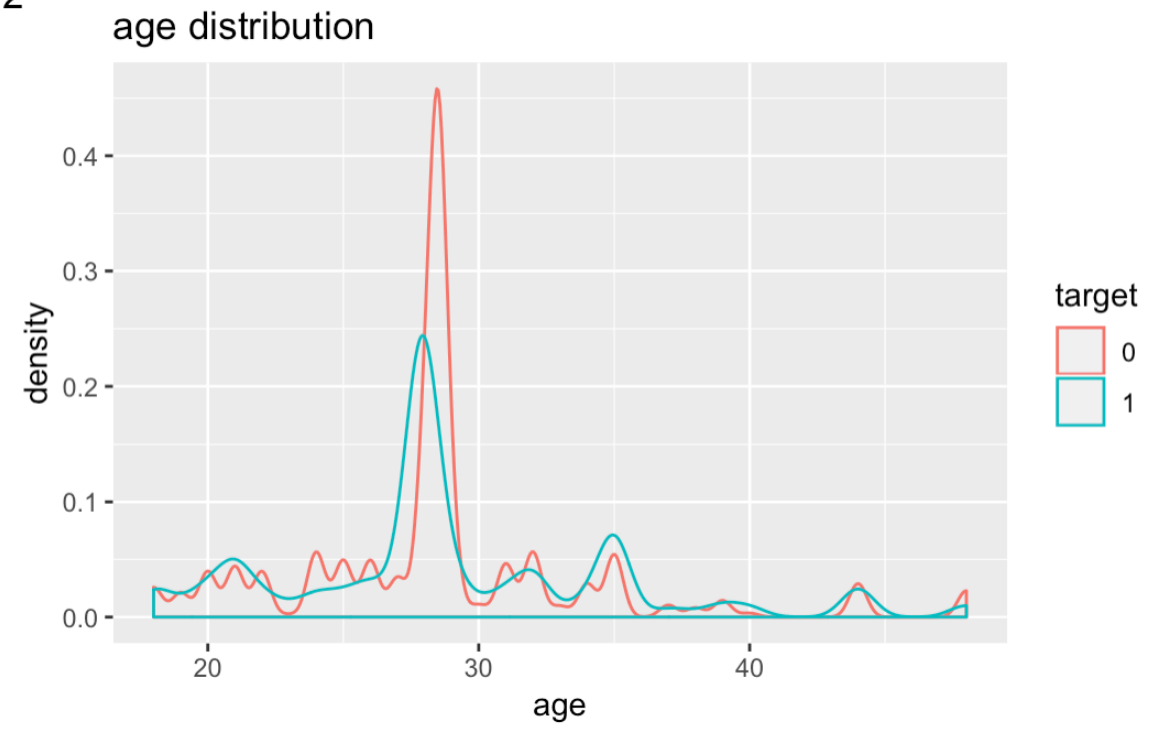
- model1: glm(target ~ 1 + msno + log(song_played_times) + scale(song_length) +log(artist_playnum) + genre_ids + n_genre + scale(bd) + source_system_tab , data = members_song_train_join, family = binomial(link = “logit"))

After fitting the model, it has a warning:"glm.fit: fitted probabilities numerically 0 or 1 occurred", which means that the model overfitted, or there is some variable significantly determined the response. In order to fix it, I drop some features and fitted model2.

2.2.2 Model2(Imporoved version of Model1)

Because "n_genre" is estimated as NA in model1, which means that it has co-linear relationship with other variables. Considering that "genre_ids" also is a predictor and contains overlapping information with "n_genre"(number of genre for a song). So I first drop this feature.

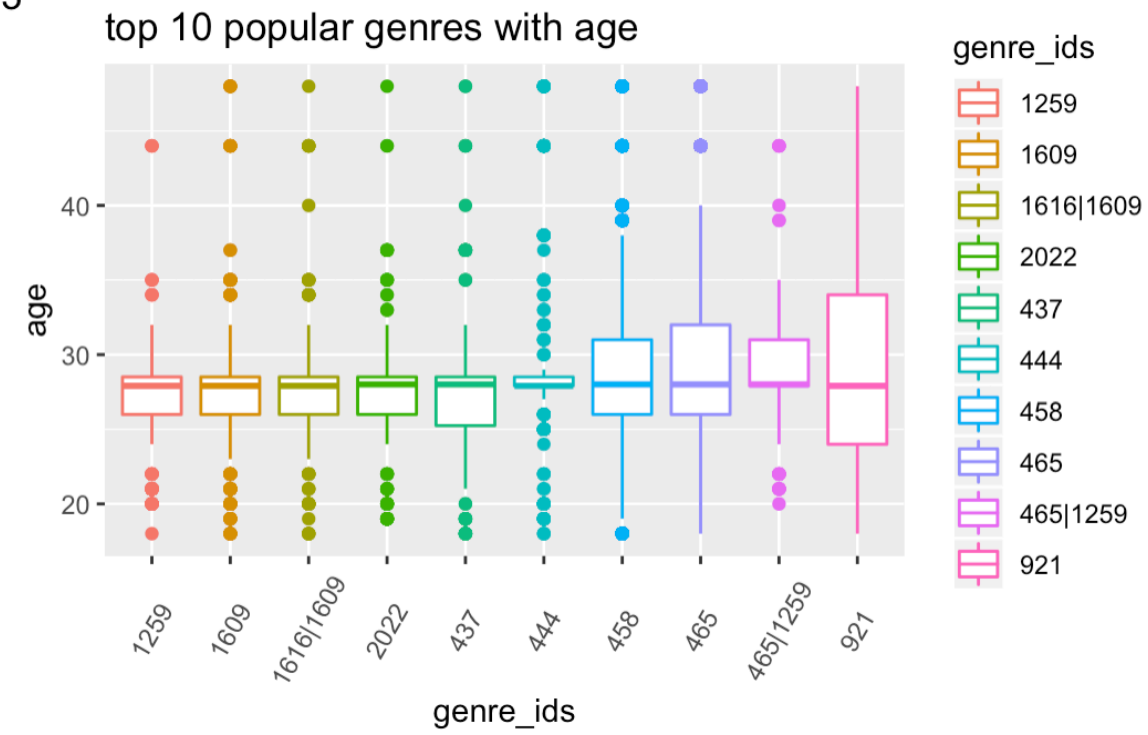
fig.12



- *From the age distribution, we can see that the probability of listening repeatedly is not linear related with age, because the red line and green line have many intersections and the upper line has a bigger probability of listening songs repeatedly. Though I tried several polynomial format for age, it still have the same warning.*

- *Then I try to figure out if there is co-linear relationship between age and other variables.*

fig.13



From this box plot we can see that different genre_ids have different scale of age of main audience. We can say that there is correlation ship between genre_ids and age/.

Considering the above 2 aspects(not linear relationship with target and co-linear with genren_idsw), I drop "bd"(age) feature, and build model2.

- Model2: glm(target ~ 1 + msno + log(song_played_times) + scale(song_length) +log(artist_playnum) + genre_ids + source_system_tab , data = members_song_train_join, family = binomial(link = "logit"))

2.2.3 Model3(Multilevel Logistic regression)

From the EDA part we can know that users have different preference. So I choose "msno"(user_id) as a group variable, building a intercept-varying model with “song_played_times","song_length","artist_playnum", "source_system_tab" as predictors.(add other variables will lead the model to non-converging.)

- Model3: glmer(target ~ (1|msno) + log(song_played_times) + scale(song_length) + scale(artist_playnum) + source_system_tab, data = members_song_train_join, family = binomial(link = "logit"))

3. Result

3.1 Model choice

3.1.1 Model2

```
msnoytPqXd07JtgbI0FiHSF21qN496xKpJYcJ8NeFu/lWgQ=  < 2e-16 ***
msnoZgPrU8gMY28iwnBPqRzDpC2K2xYytky/rsjOpSO/1Ys= 8.46e-05 ***
msnozjUhtBtcz9KbzOE+BCD5i33Fmo3z3TH2Pa430/klPec= 0.000974 ***
msnoZx4ZsWH4LUoOTadvbSKold2ftpBETVv9gAdsc2fU8KE= 0.019562 *
msnozxMVg3l0frfflcndzaKZwb9qhEXqzpouqIdwOwiPRuA= 0.106444
log(song_played_times)                            < 2e-16 ***
scale(song_length)                               0.779162
log(artist_playnum)                              0.987669
genre_ids1047                                    0.515184
genre_ids1152                                    0.006381 **


source_system_tabexplore                        3.49e-06 ***
source_system_tablisten with                    0.955358
source_system_tabmy library                      < 2e-16 ***
source_system_tabnotification                   0.209723
source_system_tabradio                           < 2e-16 ***
source_system_tabsearch                         0.002031 **
```

- song_played_times, several levels in source_system_tab("explore","library","radio","search"), several levels in genre_ids, and several levels in msno(user_id) are significant.

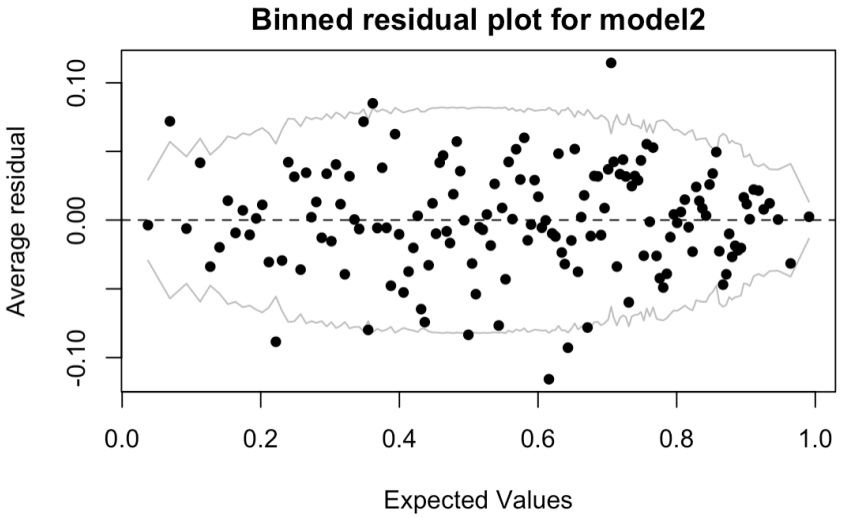
- log(song_played_times) has coefficient 0.217, which is consist with EDA result--songs played more times are more likely to be listened repeatedly.

- scale(song_length) has coefficient -0.0046. it means longer songs have lower probability to be listened repeatedly. which is not consist with EDA result.

- log(artist_playnum) has coefficient 0.00016, which is consist with EDA result--songs whose artist is played more times are more likely to be listened repeatedly.

- Null deviance: 30214 on 22115 degrees of freedom.
Residual deviance: 24760 on 21937 degrees of freedom

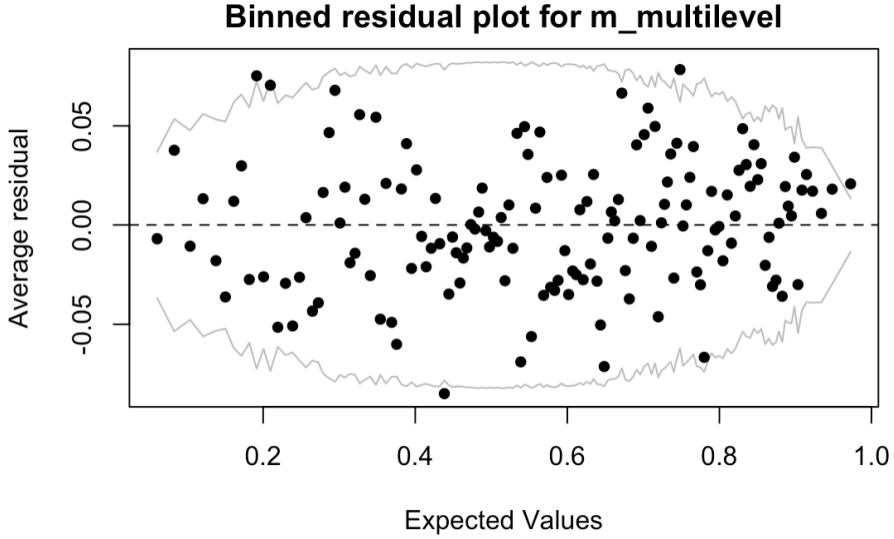
- AIC: 25116



3.1.2 Model3(Multilevel)

- AIC = 25370, deviance = 25348. both bigger than model2.

- The residual plot is better than model2, cause points beyond the CI is less.



3.1.3 Comparison

- From AIC aspect, model2 is better cause it has the least AIC,
- From deviance aspect, also model2 is better cause it has the least residual deviance.
- From residual plot, multilevel model looks better.
- We need to use test set to see which model has better classification ability.

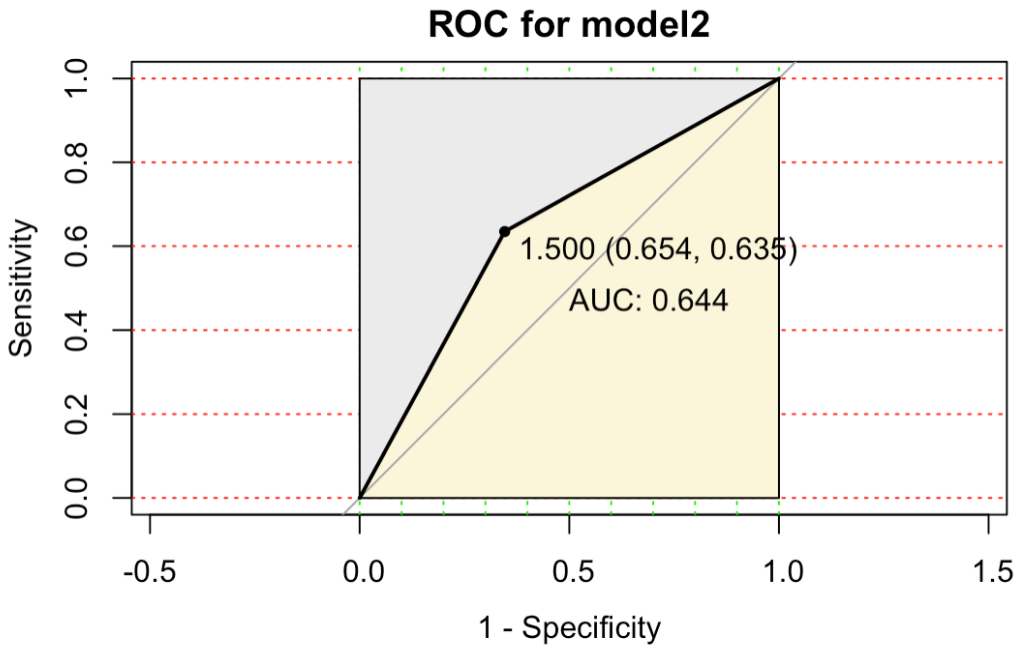
3.2 Model checking

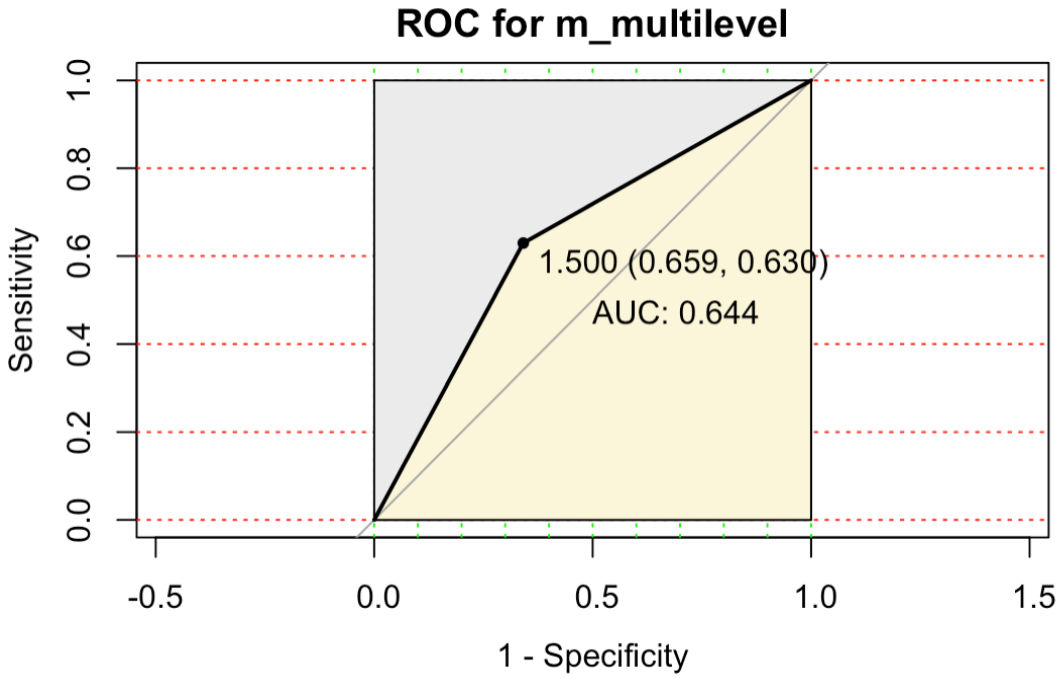
Model2

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2312 1044
         1 1223 1815

               Accuracy : 0.6454
                 95% CI : (0.6336, 0.6572)
    No Information Rate : 0.5529
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2871
 Mcnemar's Test P-Value : 0.0001851

            Sensitivity : 0.6540
            Specificity : 0.6348
         Pos Pred Value : 0.6889
         Neg Pred Value : 0.5974
             Prevalence : 0.5529
         Detection Rate : 0.3616
   Detection Prevalence : 0.5249
      Balanced Accuracy : 0.6444

       'Positive' Class : 0
```

Multilevel model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2329 1059
         1 1206 1800

               Accuracy : 0.6458
                 95% CI : (0.6339, 0.6575)
    No Information Rate : 0.5529
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.287
 Mcnemar's Test P-Value : 0.002157

            Sensitivity : 0.6588
            Specificity : 0.6296
         Pos Pred Value : 0.6874
         Neg Pred Value : 0.5988
             Prevalence : 0.5529
         Detection Rate : 0.3642
   Detection Prevalence : 0.5299
      Balanced Accuracy : 0.6442

       'Positive' Class : 0
```





- *From the ROC plots and confusion matrix, we can see that ,model2 and multilevel model is similar. Both model have AUC = 0.644.*

- *The feature "genre_ids" seems no contribution to the classification result.*

3.3 Interpretation

Because the two models do not have significant different result, we just interpret the multilevel model.

```
> display(m_multilevel)
glmer(formula = target ~ (1 | msno) + log(song_played_times) +
    scale(song_length) + scale(artist_playnum) + source_system_tab,
    data = members_song_train_join, family = binomial(link = "logit"))
                             coef.est coef.se
(Intercept)                  -1.45     0.12
log(song_played_times)        0.22     0.01
scale(song_length)           -0.01     0.02
scale(artist_playnum)         0.02     0.02
source_system_tabexplore      0.53     0.12
source_system_tablisten with -0.10     0.12
source_system_tabmy library   0.88     0.05
source_system_tabnotification -1.41     1.12
source_system_tabradio       -0.97     0.10
source_system_tabsearch       0.23     0.07

Error terms:
 Groups   Name        Std.Dev.
 msno     (Intercept) 0.90
 Residual             1.00
---
number of obs: 22116, groups: msno, 100
AIC = 25370, DIC = 24479.4
deviance = 24913.7
> ranef(m_multilevel)
$msno
                                             (Intercept)
/7IOtOs814XN7f6hlp1snYz58A1lCRC72kyhgXiSpIs=  0.5187436957
/mmecbnAd2Jrji3aonisktBko1/o+F68GqpV8X4gHiI=  0.1495023105
0E1vRvRhsSRv2byAje12xBQrr0X6NXc2R0HwjXM7AgQ=  1.0293153461
0LXSvAth90PZrowvvfSMiaZyVv35O/QXfF7/yMNuTz4= -0.8070094911
12UOkicUafP0kVfVgsllYVxpbfC0U8SgjZZMfrRMbZE=  0.1251238130
1MQUOGzTf4vCYXeOJ8vpH5gArKQvibdhOl6R6oSTLu8=  0.7237250457
25+hNlvrWoyuBaSrDJKkPj3DbZ98DAPRKoYA/hpfB6Y= -0.2867078492
2EGjY0p6GOUeQA9hHJSHbg30cN6h41OT5Is+jKJ3I1E= -0.4799359560
2F5MLrkIvrmy0RcQkekTvL1WlKGAHcKo4RaIVuCGND4= -2.4116257768
2oMc1XmEcIscvD2HgXrMcRQsaLcjhYKmPHx+TVEQK64=  0.7332079271
4aRFGkKzGUpqTbjojytAtRQNdx0egL02hjJhSqq0+NA=  0.7650530512
4DlS5LiANEKlMTQD08dakyV1JWwPYI/n1wyjx4ImjPc=  1.3202664512
4i0+qXTgsJ3Sm0s1HSHhVyGCtQxLQkcT+6G2SFiZ69M=  0.1433500228
4I3Pj3ogyd9RY0czfUOmI1pVgBbuoPXDRtbK9tsrRX0= -0.3776209409
5HRHyCakuX1hYjqQXnkGna1GX82Jo/HeM6Q1hRitNzs=  0.8860370599
5Nj4i7Wm7cC8xWtorWMyQf4r56X4Uei23JoZGR49xcU=  0.8015575115
6mTc/TeVJvQPEXzdfJSAJrh3tJ06bvD56POHzab//kc= -1.3321241926
79F8KPd1OjQn897zSEC9J9tSamY4n4E2YUxNPtJzJCs= -0.7423789931
7BwySWmWT8Ko93cgHzsizNAu2Fjpfv7gTMeGeMOy0FA= -0.5298863777
7JD/VtQIqctHF4sKBxNixuGLJ9EdGALD1hGys7wprMM=  0.9154112810
7Ymumc9vXdfuySWLG6QmCna7CXhwphtuBzFbIK4nm/M= -1.7550481340
82ezDb0HrtBGt4SnXGeMjwecSRUj4hVv6c9XDkiG45U= -0.5036641898
9gg2+52WDkcHhIJXaIkgOgYd6IhfImHVt/oD8lV7518= -1.1012356298
aAHpjH/HMTngQuoWR95lesCj/9DFHW/LmeHNxVRVXKs= -0.2490502006
AH9f0PqTy2M0r3YzQAW0XQW3AcApFesi+j3Y9ls9iIY= -0.4314534673
aSDzlImvSIf3cZQXOV2nIej5UwQxcKKwQuXJgmh2nLU= -0.1719898691
B0AoDUAjURMAjU0Mv7eB3psQxmJsvwY2/l30WR3c0fE= -0.8164097625
bdWQGPbkr5PFot+KUGc4Rs/G6gxrd4IFNvby0YDJ7Qg= -0.3746733883
```

- Different users have different intercepts. With bigger intercept, the log odds of this song being listened repeatedly by this user is bigger.
- With song_played_times increasing e times, the log odds of a song listened repeatedly increases 0.22
- With song_length increasing by sd(song_length), the log odds of a song listened repeatedly decreases 0.01.
- With artist_playnum increasing by sd(artist_playnum), the long odds of a song listened repeatedly increases 0.02.
- Songs played on different “source_system_tab” have different probabilities to be listened repeatedly. Songs played on “my library” has the biggest repeated log odds, while songs played on “radio” has the smallest repeated log odds.

4. Discussion

4.1 Implication

- In common sense, the time a song played represents the popularity of this song. And it's reasonable for a popular song to be listened repeatedly. From the result of the multilevel model, we can see the more times a song played, the more probable it will be listened repeatedly, which is consist to our common sense. So we can recommend popular songs to users.
- Song length have little influence on whether the song will be listened repeatedly or not. Longer songs are slightly less likely to be listened repeatedly. As a result, though length is not a main aspect, songs with too long length should not be significantly recommended.
- Source tab "my library" is a group including all songs this user has listened in the past. So it is reasonable that the users prefer to listen to songs which has been listened before for more times.

4.2 Limitation

- New levels for categorical variables: Linear regression is sensible to new levels not occurring in fitting data, but this situation is common in real life. For example, I want to know if a new user(without history data but only demographic data) will listen to a song repeatedly, but I

don't have the user_id in my data fitting the model, so I cannot do predict by linear regression. But in real situation, there are always new users and new songs occur.

- Missing values: For one observation, if only one variable has missing value, the whole observation need to be removed, or using other values(like mean) to replace the NA, both will decrease the accuracy of the model.

- Calculation: With categorical variables including a lot of levels, the linear regression is always lack of memory to calculate(Though the observations are only around 20000). So some features like "lyricist" and "artist" could not add into the model.

4.3 Future direction

Using machine learning algorithm like XGboost to figure out the missing value and new levels problems.

5. Acknowledgement

I would like to express my thanks to Professor Masanao for his assistance and patience with me and my model. And also thanks my friends for giving me many ideas on plots.