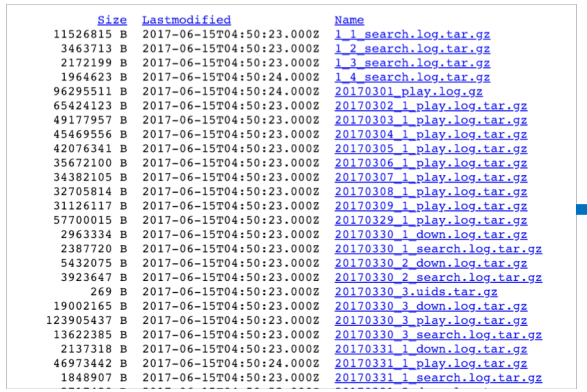
Churn prediction and recommender building on music streaming platform

Musicbox project report

Fan Yang

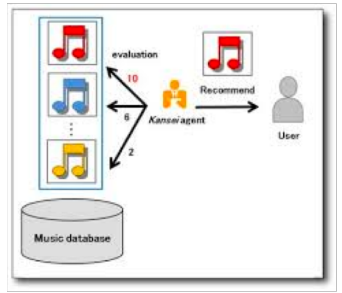
Project summary

- Musicbox is an anonymous music streaming platform.
- 392 files from Musicbox were downloaded by using Python Rerequests library.
- Each raw file contains daily user information, such as play activity, download history and search history.
- All files represents 200K users' activities for 44 days as well as basic information of 3 million songs.
- Aim1: extract features from user behavior and build churn prediction model
- Aim2: generate implicit rating from user past activity and build recommender system



Size	Lastmodified	Name
11526815 B	2017-06-15T04:50:23.000Z	1_1_search.log.tar.gz
3463713 B	2017-06-15T04:50:23.000Z	1_2_search.log.tar.gz
2172199 B	2017-06-15T04:50:23.000Z	1_3_search.log.tar.gz
1964623 B	2017-06-15T04:50:24.000Z	1_4_search.log.tar.gz
96295511 B	2017-06-15T04:50:24.000Z	20170301_play.log.gz
65424123 B	2017-06-15T04:50:23.000Z	20170302_1_play.log.tar.gz
49177957 B	2017-06-15T04:50:23.000Z	20170303_1_play.log.tar.gz
45469556 B	2017-06-15T04:50:23.000Z	20170304_1_play.log.tar.gz
42076341 B	2017-06-15T04:50:23.000Z	20170305_1_play.log.tar.gz
35672100 B	2017-06-15T04:50:23.000Z	20170306_1_play.log.tar.gz
34382105 B	2017-06-15T04:50:23.000Z	20170307_1_play.log.tar.gz
32705814 B	2017-06-15T04:50:23.000Z	20170308_1_play.log.tar.gz
31126117 B	2017-06-15T04:50:23.000Z	20170309_1_play.log.tar.gz
57700015 B	2017-06-15T04:50:23.000Z	20170329_1_play.log.tar.gz
2963334 B	2017-06-15T04:50:23.000Z	20170330_1_down.log.tar.gz
2387720 B	2017-06-15T04:50:23.000Z	20170330_1_search.log.tar.gz
5432075 B	2017-06-15T04:50:23.000Z	20170330_2_down.log.tar.gz
3923647 B	2017-06-15T04:50:23.000Z	20170330_2_search.log.tar.gz
269 B	2017-06-15T04:50:23.000Z	20170330_3.uids.tar.gz
19002165 B	2017-06-15T04:50:23.000Z	20170330_3_down.log.tar.gz
123905437 B	2017-06-15T04:50:23.000Z	20170330_3_play.log.tar.gz
13622385 B	2017-06-15T04:50:23.000Z	20170330_3_search.log.tar.gz
2137318 B	2017-06-15T04:50:23.000Z	20170331_1_down.log.tar.gz
46973442 B	2017-06-15T04:50:24.000Z	20170331_1_play.log.tar.gz
1848907 B	2017-06-15T04:50:23.000Z	20170331_1_search.log.tar.gz

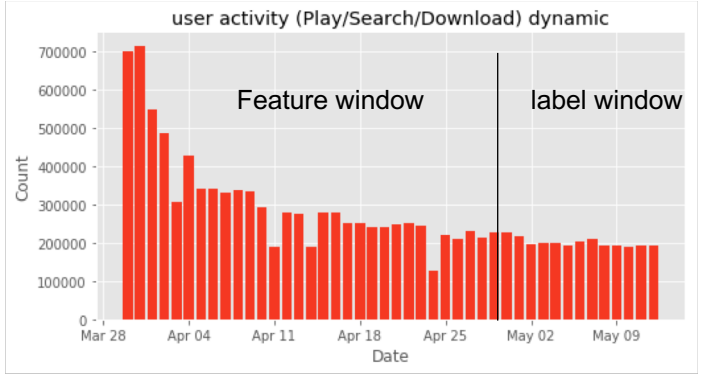
uid	total_play	play_percentage	freq_P_last_1	freq_P_last_3	freq_P_last_7	freq_P_last_14	freq_P_last_21
126941437	42975.0	0.620166	0	0	19	28	29
132952490	42539.0	0.783768	0	0	0	121	188
151294213	600.0	0.259291	0	0	0	5	5
163538558	893.0	0.654692	1	1	1	2	4
167328646	3231.0	0.936793	0	0	0	0	0



Part 1: churn prediction workflow

1. Explore and validate three type of user behavior(play, download, search) in 44 days.

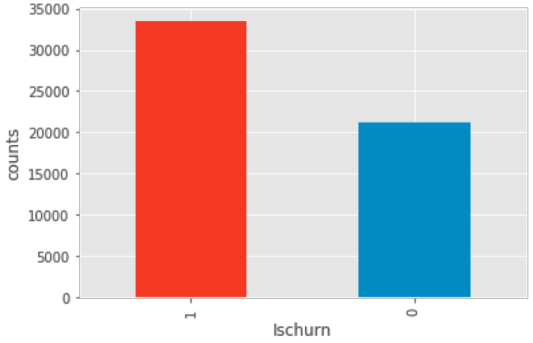
2. Identify churn user who are active in one month and became inactive during following two weeks.




```
label window: 2017-04-29 ~ 2017-05-12 days: 14
feature window: 2017-03-30 ~ 2017-04-28 days: 30
```

3. Generate features that represent frequency of user activity, music playing preference.

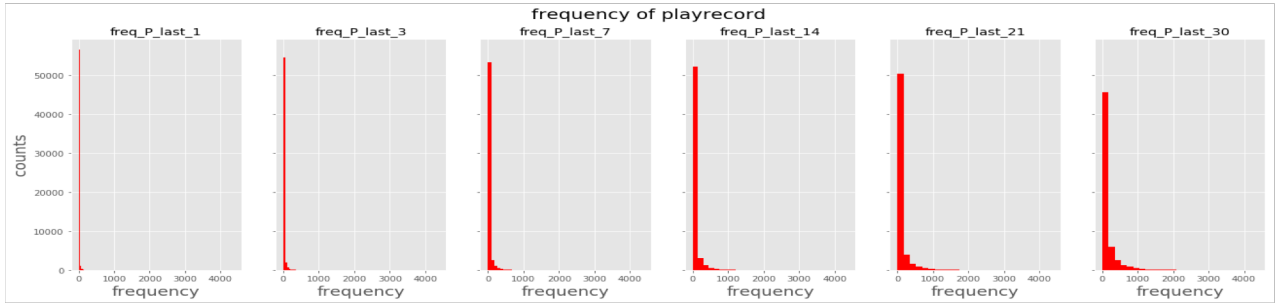
4. Build machine learning model for churn prediction



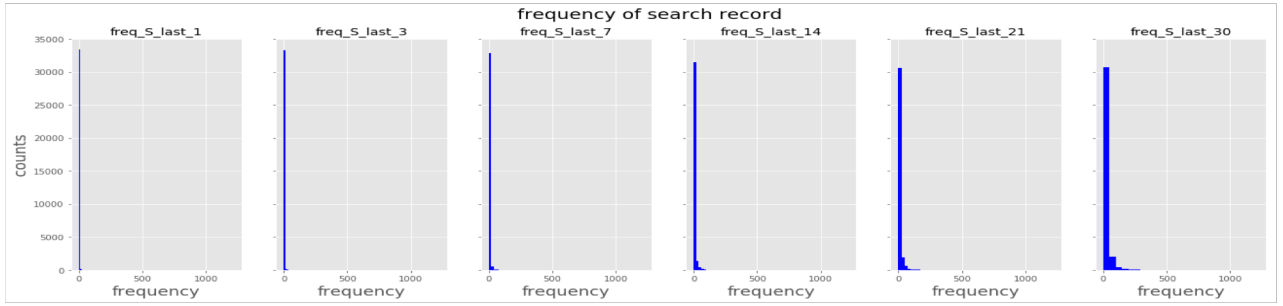
Generate frequency features of user activity

Count frequency of activities (play, search, download) in the different time frame (1, 3, 7, 14, 21, 30 days)

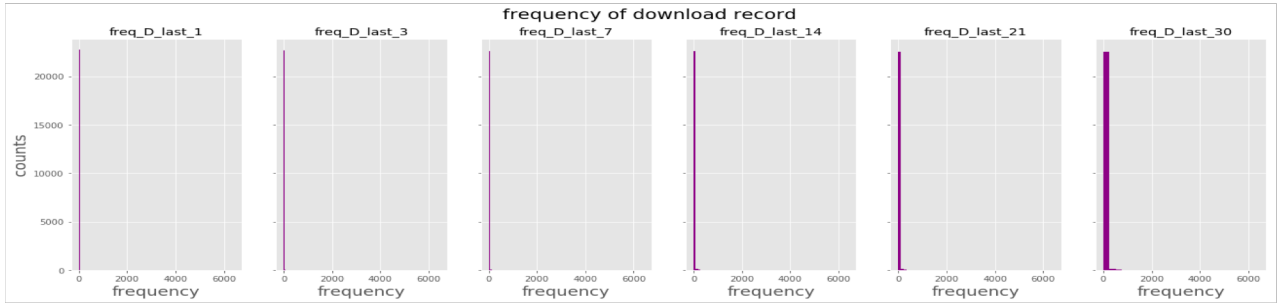
play



search



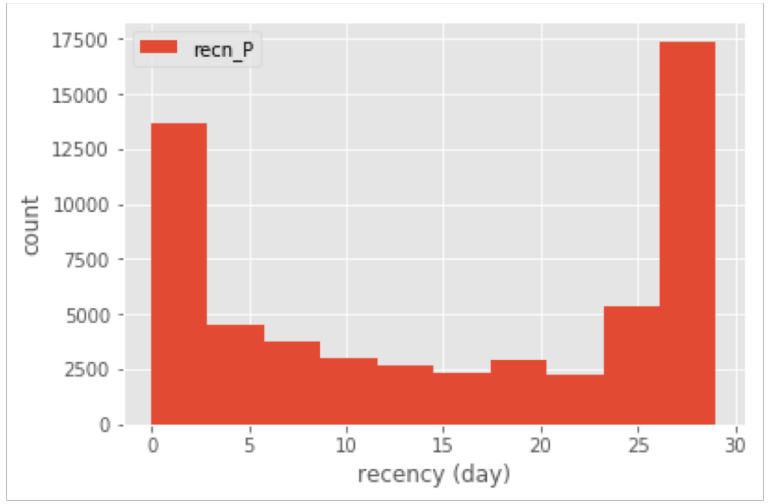
download



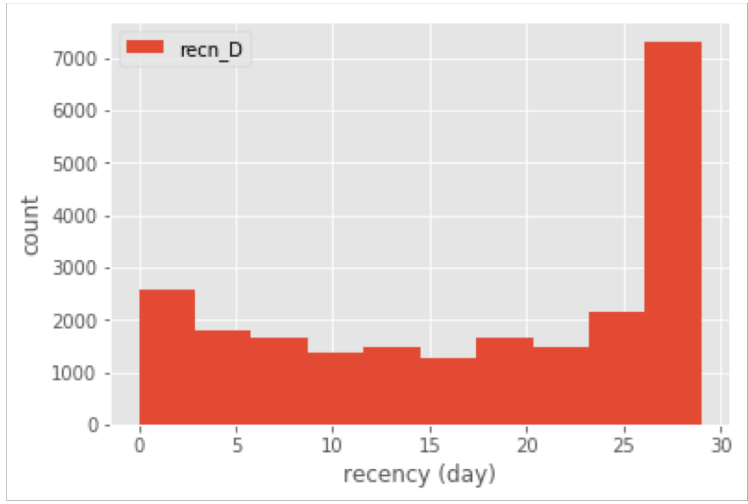
Generate recency features of user activity

Recency: number of days since user's last activity

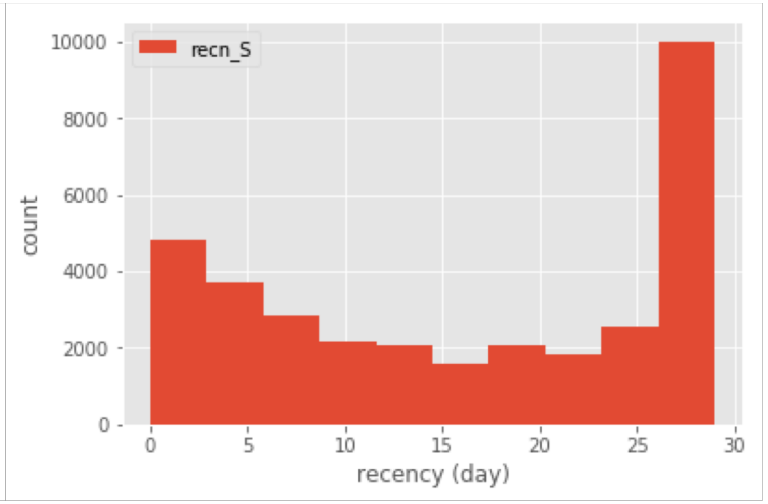
play



search



download



Generating playing-related features

1. Remove invalid ‘uid’ by using filters as below:
- ‘play_time’ >0
- ‘song_length’ >0
- ‘play_time’ <= ‘song_length’
- ’uid’ who spent less than 9 hour on daily music streaming (9 h stands for 0.99 percentile)

summary	play_time	song_length	paid_flag
count	8237117	8239170	8241922
mean	2605.2293294878123	246.72994247903623	0.0
stddev	1068882.011072512	371.94926053769467	0.0
min	-0.009905762	-1	0
max	nan	999	0

2. play_percentage = total_play ÷ total_song_length

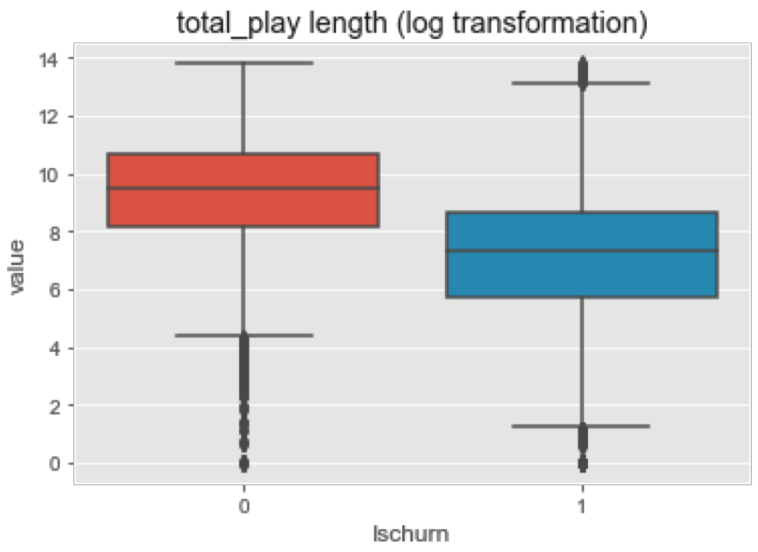
summary	uid	total_play	total_song_length	play_percentage
count	54624	54624	54624	54624
mean	1.6737449265916082E8	20991.262155387678	30041.829664219626	3.359217196436256
stddev	1.0730956346794482E7	57048.310788572075	71043.28357420773	46.008747649519165
min	100071797	0.0	2.0	0.0
max	99581051	975716.0	2734161.0	999.8111888111888

3. Group user’s device

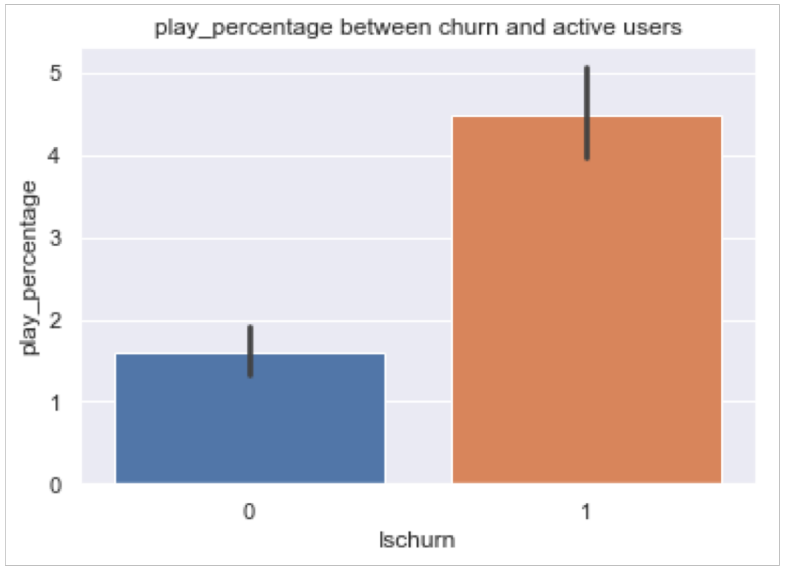
device	count
mc	2
ar	50626
ip	7342

Effects of playing-related features on churn/active user distribution

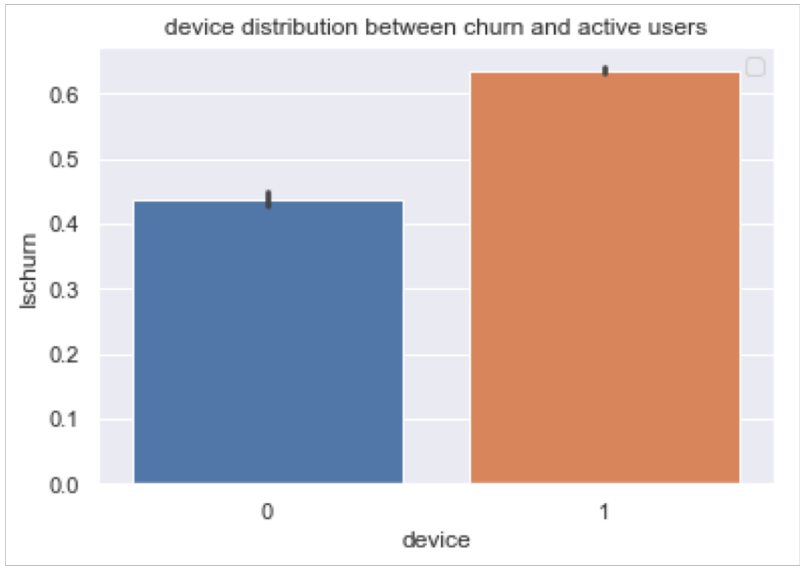
- ‘total_play_length’: the total time of playing per user
- ‘play_percentage’: the average of ‘play_length’ / ‘song_length’
- ‘device’: 0 standards for iPhone or mac, 1 standards for android phone



Churn user spent less time in playing songs



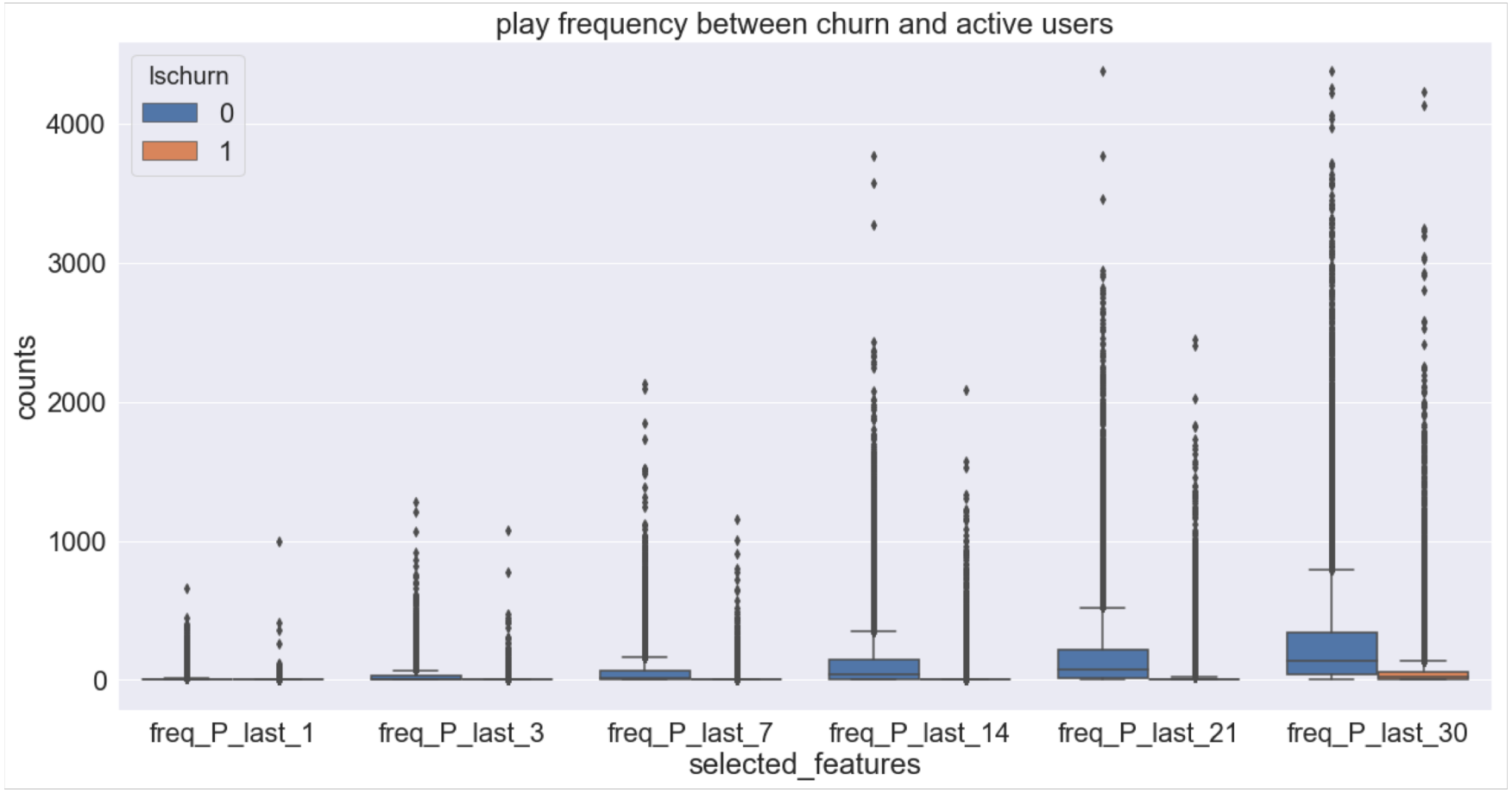
Churn user more likely to repeat playing the same song



Android users more likely to churn

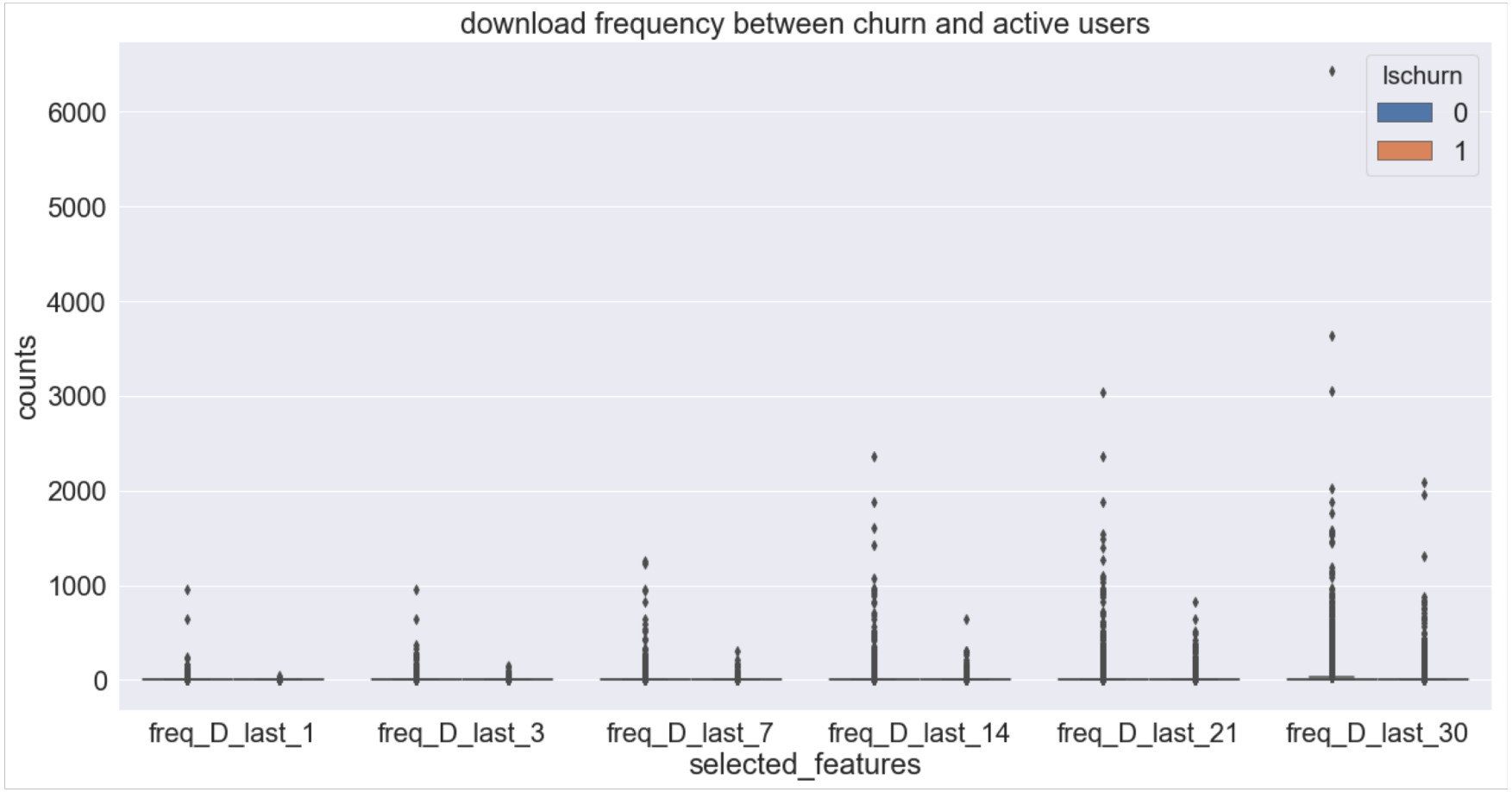
More details are include in ipynb

Effects of playing frequency features on churn/active user distribution



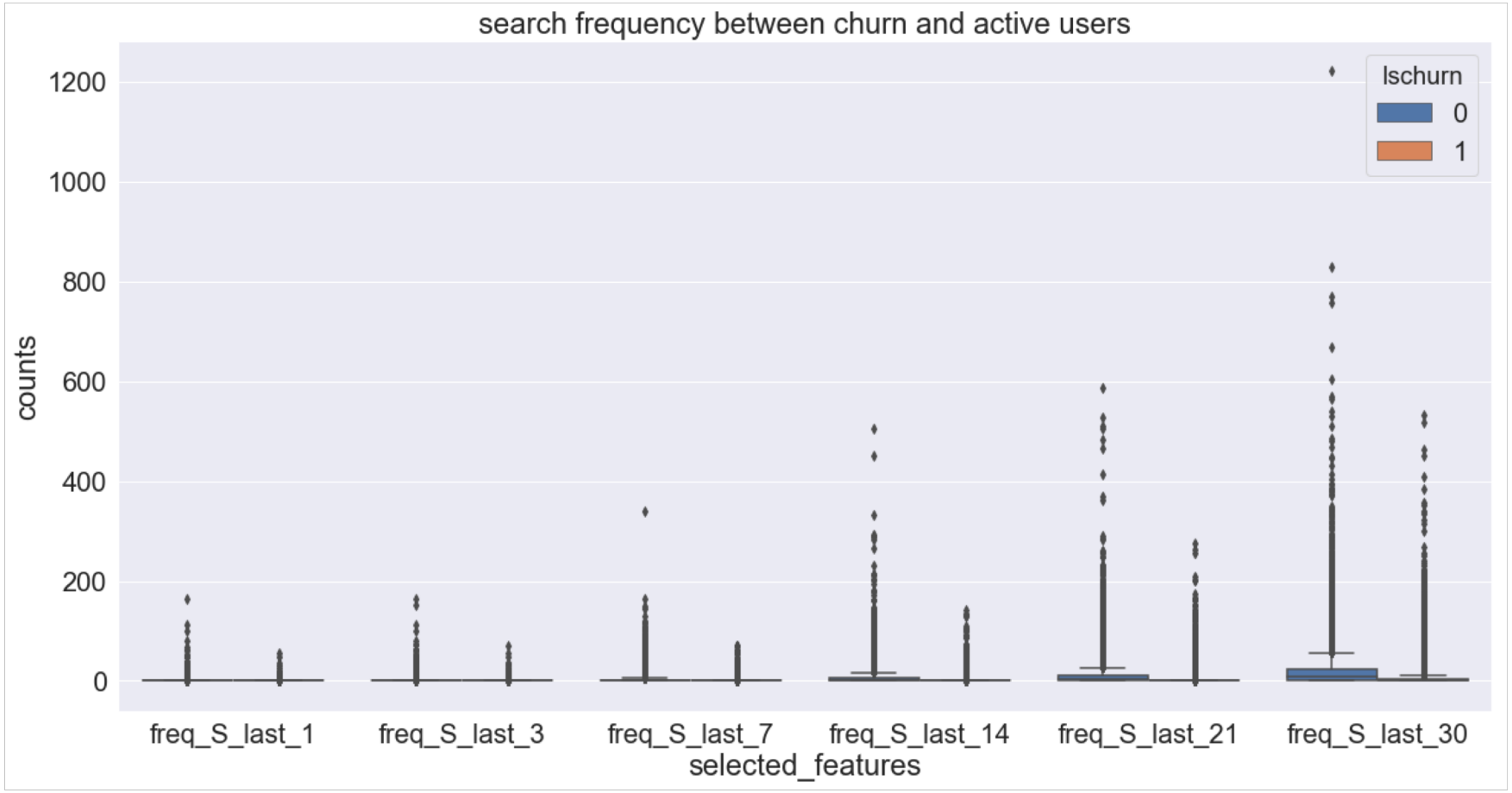
Churn user less frequently play songs in the past 30 days

Effects of download frequency features on churn/active user distribution



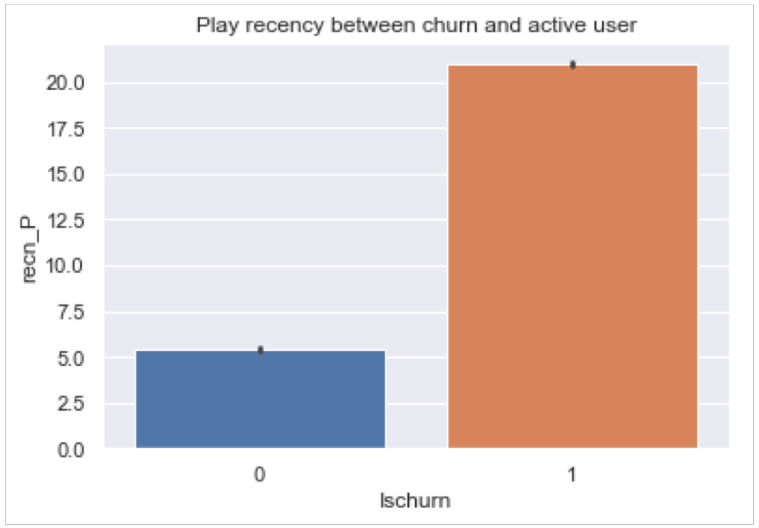
It is hard to conclude whether download frequency contribute to churn/active distribution

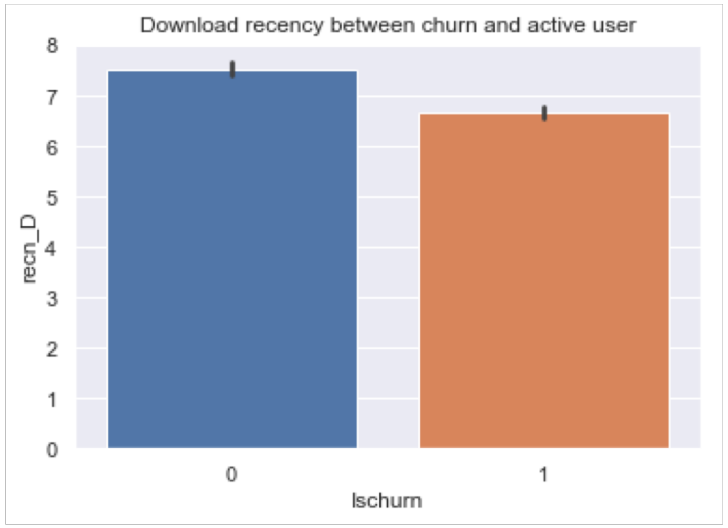
Effects of search frequency features on churn/active user distribution

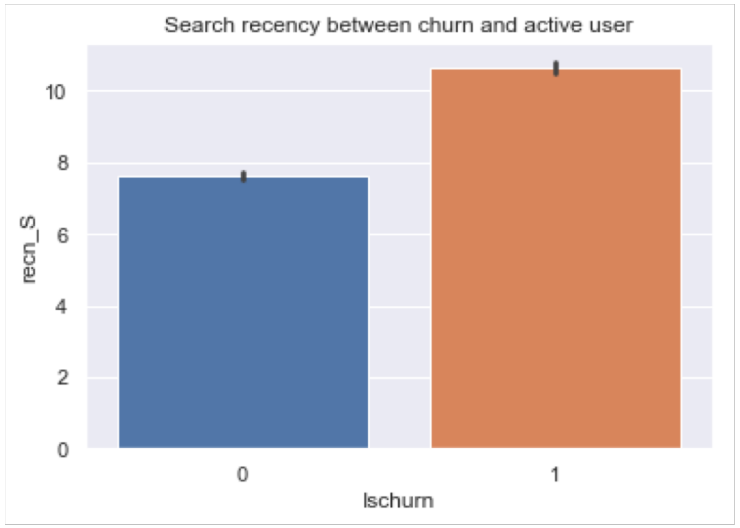


Churn user less frequently search songs in the past 30 days

Effects of recency-related features on churn/active user distribution



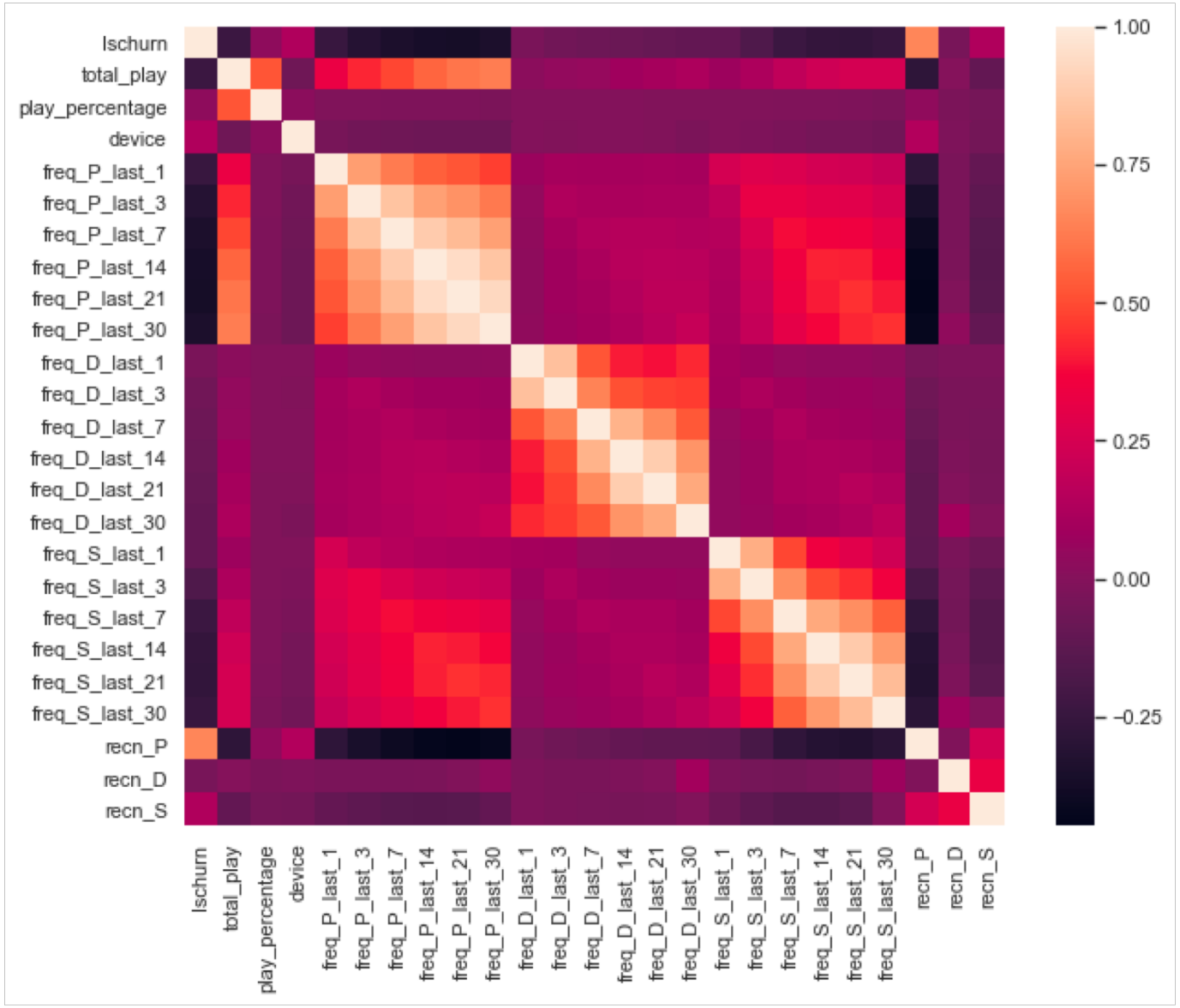




- Number of days since the last play, download and search

- Churn users show much higher number of play recency and search recency than active users.

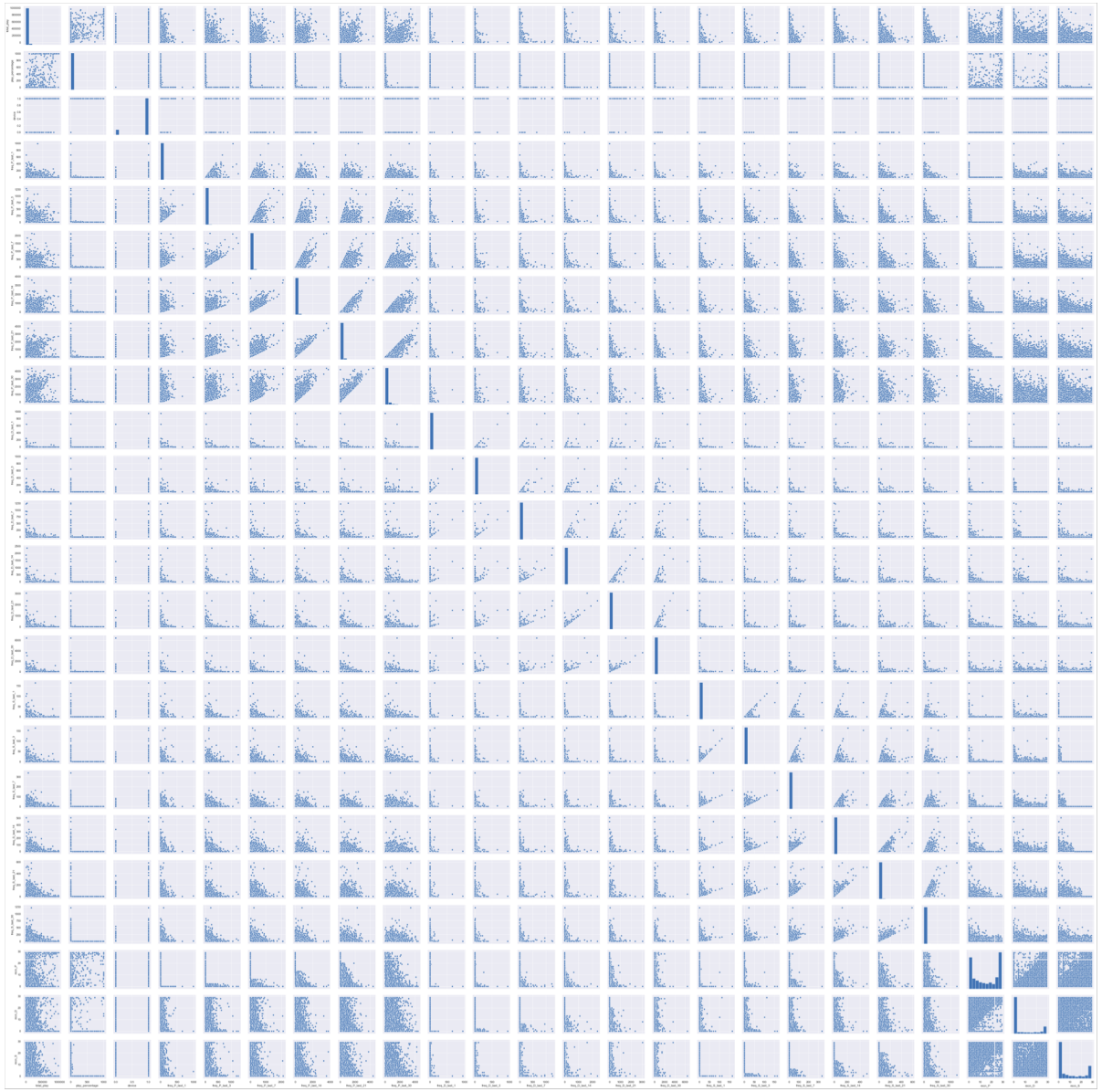
Visualizing correlations between Ischurn label and features



Top 15 churn-correlated features

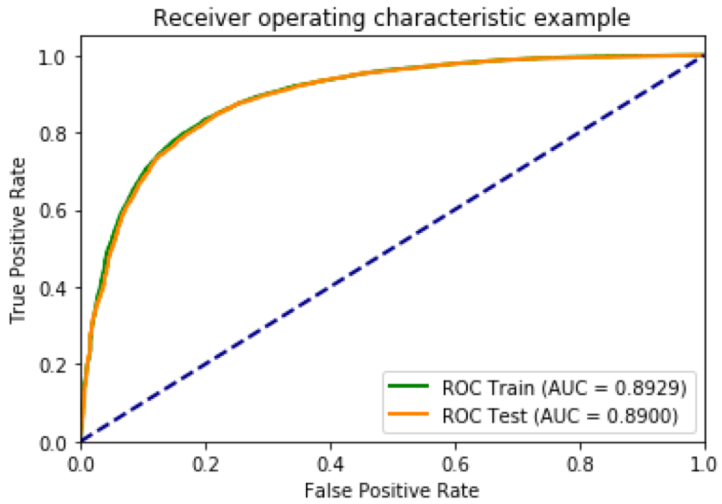
	index	corr_value
0	Ischurn	1.000000
1	recn_P	0.660198
2	device	0.135269
3	recn_S	0.130623
4	play_percentage	0.030604
5	freq_D_last_1	-0.031082
6	recn_D	-0.039212
7	freq_D_last_3	-0.057987
8	freq_D_last_7	-0.069873
9	freq_D_last_14	-0.083987
10	freq_D_last_21	-0.095234
11	freq_S_last_1	-0.102361
12	freq_D_last_30	-0.103382
13	freq_S_last_3	-0.167295
14	total_play	-0.233646
15	freq_S_last_7	-0.234391

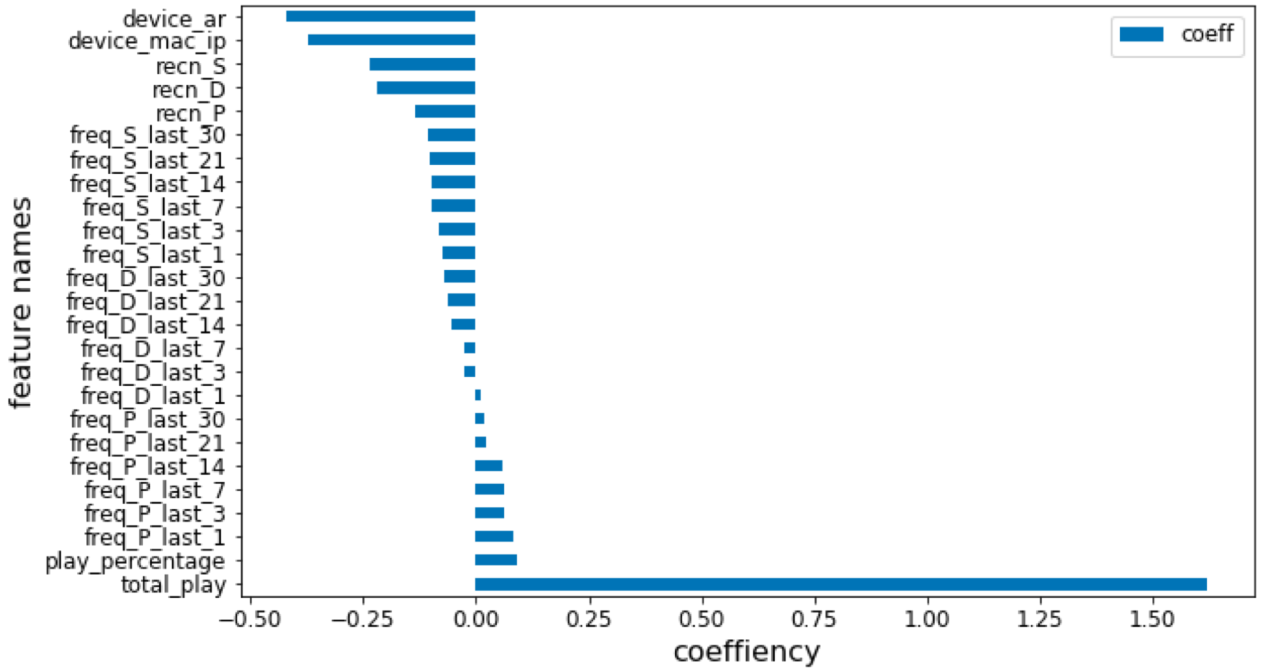
Scatter plotting Ischurn label and features



Build Logistic regression model for churn prediction

metrics	train	test
AUC	0.892886	0.889999
Accuracy	0.823354	0.821864
Precision	0.856496	0.852089
Recall	0.855181	0.855509
f1-score	0.855838	0.853795

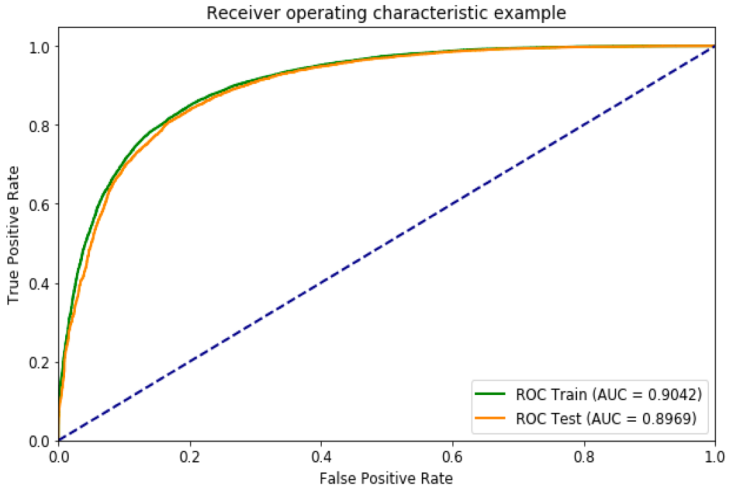


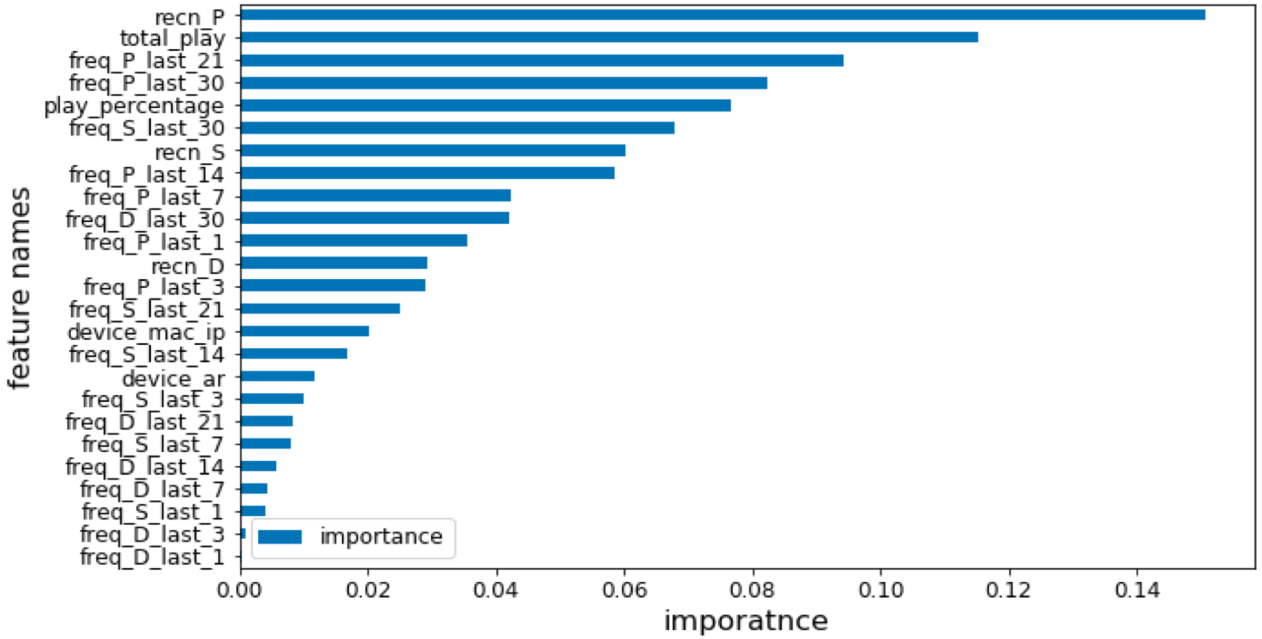


Total_play time and device type are important for churn prediction in linear model.

Build eXtreme Gradient Boosting model classifying positive review

metrics	train	test
AUC	0.904195	0.896863
Accuracy	0.834362	0.828697
Precision	0.846219	0.839757
Recall	0.891940	0.887618
f1-score	0.868478	0.863024





Playing recency and total play time are key feature for churn prediction in tree-based model

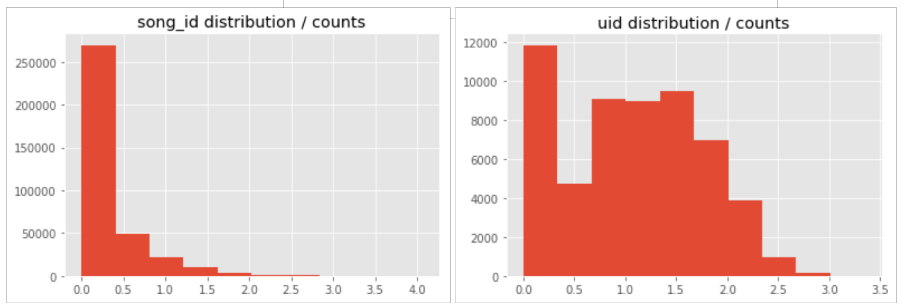
Part 2: recommend system building workflow

1. Include all data available to generate frequency features.

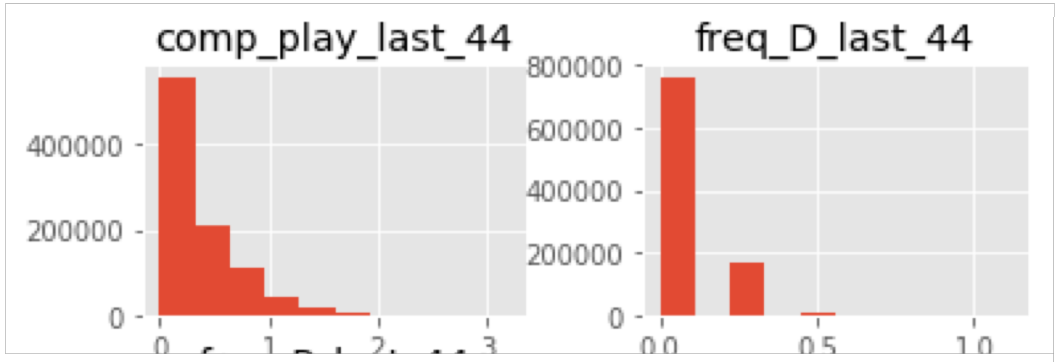
2. Validate data set and remove invalid ‘uid’ and ‘song_id’ that show abnormal pattern.

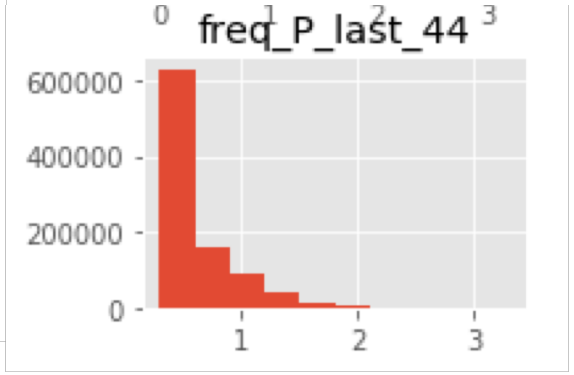
3. Select features covering longest time frame and build implicit rating.

```
+----------+----------+
|start_date|  end_date|
+----------+----------+
|2017-03-30|2017-05-12|
+----------+----------+
```

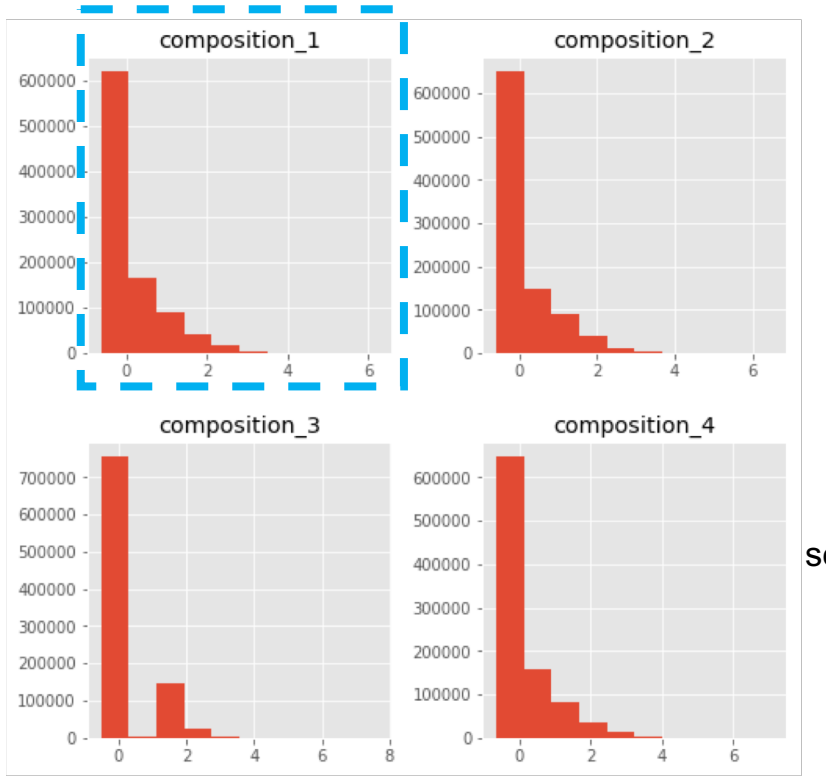


- comp_play_last_44: averaged frequency that a user play over 90% of a song in the past 44 days
- freq_P_last_44: frequency that a user play any song in the past 44 days
- freq_D_last_44: frequency that a user download any song in the past 44 days

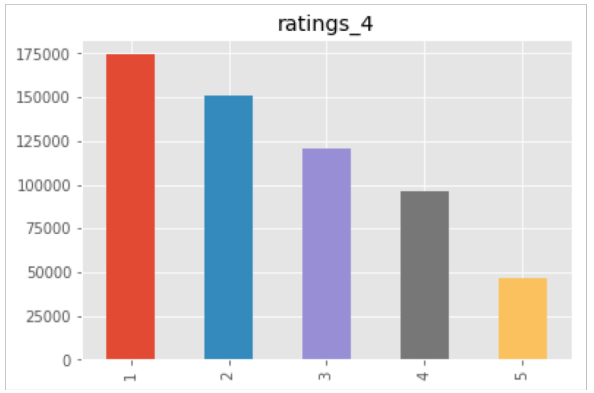




Build non-negative matrix factorization (NMF) method recommender

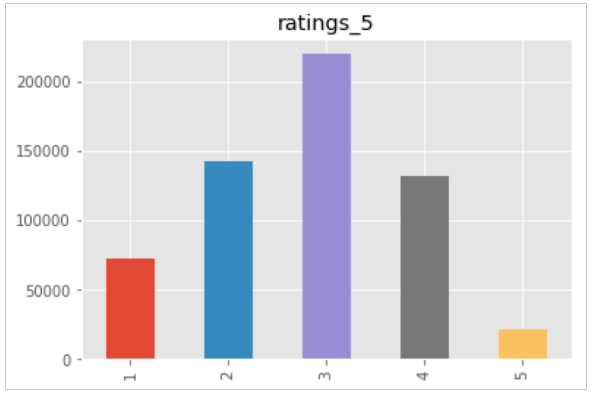


segment utility by quantile



average_abs_err
1.89

segment utility by no-skewed design



average_abs_err
2.13

1. Choose composition_1 that three features (comp_play_last_44, freq_P_last_44, freq_D_last_44) equally consist of utility
2. Generate implicit ratings
3. NMF evaluation

Build spark alternating least squares (ALS) based recommender with Root Mean Square Error (RMSE) evaluation

rating_final	count
1.0	34872
4.0	22451
3.0	176574
2.0	39329
5.0	5257

Recommendation for each user

uid	recommendations
167674030	[[508488, 4.85124...]
167979490	[[6945228, 2.3284...]
168040590	[[6945228, 3.4512...]
168736710	[[6945228, 4.4093...]
168914470	[[6945228, 4.4467...]
169006110	[[14375725, 4.618...]
168045751	[[23492088, 3.730...]
168294401	[[23657920, 3.784...]
168603631	[[6945228, 2.9099...]
168870111	[[508488, 3.38898...]
168930551	[[6945228, 4.9578...]
169018731	[[6945228, 1.8339...]
167670762	[[6945228, 3.0333...]
167735352	[[6945228, 8.3415...]
167760432	[[6945228, 4.6451...]
168236162	[[6945228, 4.8850...]
168366812	[[7148282, 5.3620...]
168838372	[[508488, 4.42948...]
168854802	[[6945228, 4.9964...]
167894223	[[15195497, 3.702...]

Recommendation for each song

song_id	recommendations
69042	[[168215501, 3.06...]
94695	[[168404412, 1.58...]
101775	[[167639478, 3.78...]
104656	[[168134566, 3.79...]
107032	[[168842861, 3.82...]
108221	[[168842861, 1.45...]
109172	[[168635054, 4.09...]
118989	[[168586329, 4.12...]
121749	[[168215501, 4.75...]
124743	[[168445833, 4.16...]
125502	[[168202866, 2.99...]
133948	[[168445833, 4.63...]
135976	[[168445833, 4.38...]
144907	[[168138422, 3.05...]
156365	[[168586329, 5.48...]
158257	[[168034451, 3.76...]
197258	[[168260599, 4.41...]
200878	[[168445833, 4.81...]
202641	[[168445833, 3.10...]
204529	[[168700215, 3.19...]

- When implicitPrefs = False, RMSE=0.89; When implicitPrefs = False, RMSE=2.83.
- It suggests that RMSE is not a perfect way to evaluate ALS-based recommender model

Build spark ALS based recommender with ranking evaluation

implicit_ratings	count
1.0	21681
4.0	16371
3.0	120856
2.0	23822
5.0	3759

Predicted recommendations by model

uid	song_id
167582087	[3247615, 703939,...]
167627297	[6660691]
167674030	[22847594, 983871...]
167683346	[6196652, 6957813...]
167697454	[930067, 102811, ...]
167746855	[3626563]
167760432	[6635281, 1072044...]
167888996	[1149606, 4117304...]
167894223	[21596231, 222862...]
167913407	[6976836]
167993496	[1147548, 6509249...]
168045751	[6408557]
168054336	[157606, 122173, ...]
168150258	[328037]
168236162	[224887, 12442906...]
168275526	[854122, 4170546,...]
168275915	[6892457]
168295039	[6651583]
168359214	[6926187, 6855384]
168363559	[3440648, 1035446...]

Observed recommendations from user past behavior

uid	song_id
167582087	[703939, 22866732...]
167627297	[6660691]
167674030	[22056152, 716376...]
167683346	[399964, 3973161,...]
167697454	[394129, 949612, ...]
167746855	[3626563]
167760432	[3314628, 1099977...]
167888996	[7141522, 1149606...]
167894223	[7095195, 706277,...]
167913407	[6976836]
167993496	[15830559, 114754...]
168045751	[6408557]
168054336	[122173, 157856, ...]
168150258	[328037]
168236162	[897464, 955482, ...]
168275526	[4441094, 4170546...]
168275915	[6892457]
168295039	[6651583]
168359214	[6926187, 6855384]
168363559	[3348840, 3440648...]

```
als = ALS(implicitPrefs=True, seed=42,userCol="uid", itemCol="song_id", ratingCol="implicit_ratings") \
    .setRank(50) \
    .setMaxIter(22) \
    .setRegParam(0.5) \
    .setAlpha(40)

model = als.fit(train)
```

```
evaluation of model_1:
metric:  0.9684
meanAveragePrecision:  0.9549
precisionAt:  0.1343
```

Project conclusions

1. This project analyze user past behavior on a music streaming platform and generate features representing frequency, recency and listening activities.
2. Logistic regress and boosting models were built to predict churn user. The AUC on test dataset reach 0.9.
3. The models identify features are important for churn prediction, such as device type and frequency and recency of playing activity.
4. Implicit ratings were generated based user past behavior to represent user's preference on different songs
5. Recommend systems were build by using NMF, spark ALS methods and their recommendation accuracy were evaluated by RMSE and ranking metrics.