Initail Attempt without Feature Engineering

June 3, 2020

In [2]:
```
import pandas as pd
import numpy as np
```

In [18]:
```
# load data
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
songs = pd.read_csv('songs.csv')
members = pd.read_csv('members.csv', parse_dates=["registration_init_time","expiratio
songs_extra_info = pd.read_csv('song_extra_info.csv')
```

In [4]:
```
train.head()
```

Out[4]:

	msno	song_id	source_system_tab	source_screen_name	source_type	target
0	FGtllVqz18RPiwJj/edr2gV78zirAiY/9SmYvia+kCg=	BBzumQNXUHKdEBOB7mAJuzok+IJA1c2Ryg/yzTF6tik=	explore	Explore	online-playlist	1
1	Xumu+NIjS6QYVxDS4/t3SawvJ7viT9hPKXmf0RtLNx8=	bhp/MpSNoqoxOIB+/l8WPqu6jldth4DIpCm3ayXnJqM=	my library	Local playlist more	local-playlist	1
2	Xumu+NIjS6QYVxDS4/t3SawvJ7viT9hPKXmf0RtLNx8=	JNWfrrC7zNN7BdMpsISKa4Mw+xVJYNnxXh3/Epw7QgY=	my library	Local playlist more	local-playlist	1
3	Xumu+NIjS6QYVxDS4/t3SawvJ7viT9hPKXmf0RtLNx8=	2A87tzfnJTSWqD7gIZHisolhe4DMdzkbd6LzO1KHjNs=	my library	Local playlist more	local-playlist	1
4	FGtllVqz18RPiwJj/edr2gV78zirAiY/9SmYvia+kCg=	3qm6XTZ6MOCU11x8FIVbAGH5l5uMkT3/ZalWG1oo2Gc=	explore	Explore	online-playlist	1

In [10]:
```
# change object to category
for col in train.columns:
    if train[col].dtype == object:
        train[col] = train[col].astype('category')
        test[col] = test[col].astype('category')
```

```
In [17]: train.info()
         print('\n')
         test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7377418 entries, 0 to 7377417
Data columns (total 6 columns):
msno                    category
song_id                 category
source_system_tab       category
source_screen_name      category
source_type             category
target                  int64
dtypes: category(5), int64(1)
memory usage: 133.8 MB


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2556790 entries, 0 to 2556789
Data columns (total 6 columns):
id                      int64
msno                    category
song_id                 category
source_system_tab       category
source_screen_name      category
source_type             category
dtypes: category(5), int64(1)
memory usage: 54.0 MB
```

```
In [12]: # Split the original train set
         # 75% as the train set, 25% as the validation set.

         index = round(len(train)*0.75) # round to nearest integer
         tr_set = train.iloc[0:index,:]
         val_set = train.iloc[index+1:,:]

         X_tr = tr_set.drop(['target'], axis=1)
         y_tr = tr_set['target'].values

         X_val = val_set.drop(['target'], axis=1)
         y_val = val_set['target'].values
```

```
In [13]: #LightGBM
         import lightgbm as lgb
         lgb_train = lgb.Dataset(X_tr, y_tr)
         lgb_val = lgb.Dataset(X_val, y_val)
```

```
In [14]: params = {
                 'objective': 'binary',
```

```
            'boosting': 'gbdt',
            'learning_rate': 0.2 ,
            'verbose': 0,
            'num_leaves': 100,
            'bagging_fraction': 0.95,
            'bagging_freq': 1,
            'bagging_seed': 1,
            'feature_fraction': 0.9,
            'feature_fraction_seed': 1,
            'max_bin': 256,
            'num_rounds': 100,
            'metric' : 'auc'
        }

    lgbm_model = lgb.train(params, train_set = lgb_train, valid_sets = lgb_val, verbose_ev
```

```
[5]         valid_0's auc: 0.640136
[10]        valid_0's auc: 0.647537
[15]        valid_0's auc: 0.653946
[20]        valid_0's auc: 0.65802
[25]        valid_0's auc: 0.661291
[30]        valid_0's auc: 0.66283
[35]        valid_0's auc: 0.663972
[40]        valid_0's auc: 0.664686
[45]        valid_0's auc: 0.665381
[50]        valid_0's auc: 0.665831
[55]        valid_0's auc: 0.666081
[60]        valid_0's auc: 0.666397
[65]        valid_0's auc: 0.666549
[70]        valid_0's auc: 0.666545
[75]        valid_0's auc: 0.666661
[80]        valid_0's auc: 0.666744
[85]        valid_0's auc: 0.666705
[90]        valid_0's auc: 0.666762
[95]        valid_0's auc: 0.666795
[100]       valid_0's auc: 0.666719
```

In [15]:
```
# predict the test set
ids = test['id'].values
X_test = test.drop(['id'], axis=1)

predictions = lgbm_model.predict(X_test)

# Writing output to csv
subm = pd.DataFrame()
subm['id'] = ids
subm['target'] = predictions
subm.to_csv('submission.csv', index=False)
```