Project 5010

June 3, 2020

#
Yang HUANG 20656570

0.1 1. Data Processing

In [21]:
```
import pandas as pd
import numpy as np
```

In [22]:
```
# load data
train0 = pd.read_csv('train.csv')
test0 = pd.read_csv('test.csv')
songs0 = pd.read_csv('songs.csv')
members0 = pd.read_csv('members.csv',parse_dates=["registration_init_time","expiratio
songs_extra_info0 = pd.read_csv('song_extra_info.csv')
```

In [23]:
```
print("train")
train0.info()
print("\n")
print("test")
test0.info()
print("\n")
print("members")
members0.info()
print("\n")
print("songs")
songs0.info()
print("\n")
print("songs_extra_info")
songs_extra_info0.info()
```

```
train
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7377418 entries, 0 to 7377417
Data columns (total 6 columns):
msno                    object
song_id                 object
source_system_tab       object
source_screen_name      object
```

```
source_type           object
target                int64
dtypes: int64(1), object(5)
memory usage: 337.7+ MB
```

```
test
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2556790 entries, 0 to 2556789
Data columns (total 6 columns):
id                    int64
msno                  object
song_id               object
source_system_tab     object
source_screen_name    object
source_type           object
dtypes: int64(1), object(5)
memory usage: 117.0+ MB
```

```
members
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34403 entries, 0 to 34402
Data columns (total 7 columns):
msno                     34403 non-null object
city                     34403 non-null int64
bd                       34403 non-null int64
gender                   14501 non-null object
registered_via           34403 non-null int64
registration_init_time   34403 non-null datetime64[ns]
expiration_date          34403 non-null datetime64[ns]
dtypes: datetime64[ns](2), int64(3), object(2)
memory usage: 1.8+ MB
```

```
songs
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2296320 entries, 0 to 2296319
Data columns (total 7 columns):
song_id        object
song_length    int64
genre_ids      object
artist_name    object
composer       object
lyricist       object
language       float64
dtypes: float64(1), int64(1), object(5)
memory usage: 122.6+ MB
```

```
songs_extra_info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2295971 entries, 0 to 2295970
Data columns (total 3 columns):
song_id    object
name       object
isrc       object
dtypes: object(3)
memory usage: 52.6+ MB
```

0.1.1 Change object to catagory

```
In [24]: def object_to_category(df):
             for col in df.columns:
                 if df[col].dtype == object:
                     df[col] = df[col].astype('category')
         object_to_category(train0)
         object_to_category(test0)
         object_to_category(members0)
         object_to_category(songs0)
         object_to_category(songs_extra_info0)

In [25]: # for temporary assignment
         train = train0
         test = test0
         songs = songs0
         members = members0
         songs_extra_info = songs_extra_info0
```

0.2 2. Explantory Data Analysis (EDA)

This part is done in another notebook seperately

0.3 3. Feature Engineering

0.3.1 members

```
In [26]: members['membership_days'] = members['expiration_date'].subtract(members['registratio
         #members['registration_year'] = members['registration_init_time'].dt.year
         #members['registration_month'] = members['registration_init_time'].dt.month
         #members['registration_day'] = members['registration_init_time'].dt.day

         #members['expiration_year'] = members['expiration_date'].dt.year
         #members['expiration_month'] = members['expiration_date'].dt.month
         #members['expiration_day'] = members['expiration_date'].dt.day
         members1 = members.drop(['registration_init_time','expiration_date'], axis=1)
```

```
In [27]: train = train.merge(members1, on='msno', how='left')
         test = test.merge(members1, on='msno', how='left')
         # train & test have no null in membership_days
```

train & test have no null in membership_days. So, it's no need to handle missing values

0.3.2 songs

```
In [28]: songs.song_length.fillna(200000, inplace=True)

         def long_song_bool(x):
             if x > 400000:
                 return 1
             return 0
         songs['long_song_bool'] = songs['song_length'].apply(long_song_bool)

In [29]: songs1 = songs[['song_id','artist_name','genre_ids','language','long_song_bool']]
         train = train.merge(songs1, on='song_id', how='left')
         test = test.merge(songs1, on='song_id', how='left')

In [30]: # Deal with the missing values
         train['language'].fillna(-100, inplace=True) # replace nan with -100
         test['language'].fillna(-100, inplace=True) # replace nan with -100
         train['language']=train['language'].astype("category")
         test['language']=test['language'].astype("category")

         train['artist_name']=train['artist_name'].cat.add_categories("no_artist").fillna('no_a
         test['artist_name']=test['artist_name'].cat.add_categories("no_artist").fillna('no_ar

         train['genre_ids']=train['genre_ids'].cat.add_categories("no_genre").fillna('no_genre
         test['genre_ids']=test['genre_ids'].cat.add_categories("no_genre").fillna('no_genre')

In [31]: #test['artist_name']=test['artist_name'].cat.add_categories("unknown").fillna('unknow
         #train[train['artist_name']=="unknown"]
         #g_without_nan = g.cat.add_categories("D").fillna("D")
         #train['artist_name']
         #len(test)
```

0.3.3 songs_extra_info

```
In [32]: songs_extra_info = songs_extra_info0
         songs_extra_info0.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2295971 entries, 0 to 2295970
Data columns (total 3 columns):
song_id    category
name       category
isrc       category
```

```
dtypes: category(3)
memory usage: 266.5 MB
```

```
In [33]: def isrc_to_year(isrc):
             if type(isrc) == str:
                 if int(isrc[5:7]) > 17:
                     return 1900 + int(isrc[5:7])
                 else:
                     return 2000 + int(isrc[5:7])
             else:
                 return np.nan

         songs_extra_info['song_year'] = songs_extra_info['isrc'].apply(isrc_to_year)
```

```
In [34]: songs_extra_info1 = songs_extra_info[["song_id","song_year"]]
         songs_extra_info1['song_year'] = songs_extra_info1['song_year'].astype("category")
         train = train.merge(songs_extra_info1, on='song_id', how='left')
         test = test.merge(songs_extra_info1, on='song_id', how='left')
```

```
/Applications/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: SettingWithCopyWa
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
```

0.3.4 train set after being merged

0.3.5 Also some feartures are adding here....

```
In [35]: def artist_count(x):
             if x == 'unknown':
                 return 0
             else:
                 return x.count('and') + x.count('+')+ x.count(',') + x.count('feat') + x.count

         train['artist_count'] = train['artist_name'].apply(artist_count).astype(np.int8)
         test['artist_count'] = test['artist_name'].apply(artist_count).astype(np.int8)
```

```
In [36]: def genre_id_count(x):
             if x == 'unknown':
                 return 0
             else:
                 return x.count('|') + 1

         train['genre_ids_count'] = train['genre_ids'].apply(genre_id_count).astype(np.int8)
         test['genre_ids_count'] = test['genre_ids'].apply(genre_id_count).astype(np.int8)
```

In [37]:
```
# Count number of times that each song was played before
temp = pd.concat([train,test])
song_played_count = temp[["song_id","target"]].groupby(["song_id"],as_index=False).cou
train = train.merge(song_played_count, on='song_id', how='left')
test = test.merge(song_played_count, on='song_id', how='left')
```

```
/Applications/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: FutureWarning: So
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.
```

In [38]:
```
# Count number of times that each artist was played before
artist_played_count = temp[["artist_name","target"]].groupby(["artist_name"],as_index=
train = train.merge(artist_played_count, on='artist_name', how='left')
test = test.merge(artist_played_count, on='artist_name', how='left')
```

In [39]:
```
# Check out the user activation
user_occurrence_count = temp[["msno","target"]].groupby(["msno"],as_index=False).count
train = train.merge(user_occurrence_count, on='msno', how='left')
test = test.merge(user_occurrence_count, on='msno', how='left')
```

In [41]:
```
# Encoding

from sklearn.preprocessing import LabelEncoder

encoding_list = ['source_system_tab', 'source_screen_name','source_type']
for i in encoding_list:
    lb = LabelEncoder()
    lb.fit(list(train[i].values) + list(test[i].values))
    train[i] = lb.transform(list(train[i].values))
    test[i] = lb.transform(list(test[i].values))
```

In [42]:
```
print("train")
train.info()
print("\n")
print("test")
test.info()
```

```
train
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7377418 entries, 0 to 7377417
Data columns (total 21 columns):
msno                    object
song_id                 object
```

```
source_system_tab        int64
source_screen_name       int64
source_type              int64
target                   int64
city                     int64
bd                       int64
gender                   category
registered_via           int64
membership_days          int64
artist_name              category
genre_ids                category
language                 category
long_song_bool           float64
song_year                category
artist_count             int8
genre_ids_count          int8
song_played_count        int64
artist_played_count      int64
user_occurrence_count    int64
dtypes: category(5), float64(1), int64(11), int8(2), object(2)
memory usage: 933.4+ MB


test
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2556790 entries, 0 to 2556789
Data columns (total 21 columns):
id                       int64
msno                     object
song_id                  object
source_system_tab        int64
source_screen_name       int64
source_type              int64
city                     int64
bd                       int64
gender                   category
registered_via           int64
membership_days          int64
artist_name              category
genre_ids                category
language                 category
long_song_bool           float64
song_year                category
artist_count             int8
genre_ids_count          int8
song_played_count        int64
artist_played_count      int64
user_occurrence_count    int64
```

```
dtypes: category(5), float64(1), int64(11), int8(2), object(2)
memory usage: 331.2+ MB
```

```
In [43]: #change into category
         object_to_category(train)
         object_to_category(test)
```

0.4 4. LightGBM

```
In [44]: # 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 [45]: #LightGBM
         import lightgbm as lgb
         lgb_train = lgb.Dataset(X_tr, y_tr)
         lgb_val = lgb.Dataset(X_val, y_val)
```

```
/Applications/anaconda3/lib/python3.7/site-packages/lightgbm/__init__.py:48: UserWarning: Start
This means that in case of installing LightGBM from PyPI via the ``pip install lightgbm`` comma
Instead of that, you need to install the OpenMP library, which is required for running LightGBM
You can install the OpenMP library by the following command: ``brew install libomp``.
  "You can install the OpenMP library by the following command: ``brew install libomp``.", User
```

```
In [46]: 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
```

```
/Applications/anaconda3/lib/python3.7/site-packages/lightgbm/engine.py:148: UserWarning: Found
  warnings.warn("Found `{}` in params. Will use it instead of argument".format(alias))
```

```
[10]        valid_0's auc: 0.660179
[20]        valid_0's auc: 0.672338
[30]        valid_0's auc: 0.67774
[40]        valid_0's auc: 0.680318
[50]        valid_0's auc: 0.681399
[60]        valid_0's auc: 0.681865
[70]        valid_0's auc: 0.682131
[80]        valid_0's auc: 0.682737
[90]        valid_0's auc: 0.683226
[100]        valid_0's auc: 0.683528
```

In [47]:
```
# 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('submission4.csv', index=False)
```