```
import pandas as pd
from sklearn import model_selection
# import lightgbm as lgb
import os
import sys
import shutil
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
# from catboost import CatBoostClassifier
!pip install lightfm

from lightfm import LightFM
import scipy.sparse as sp



!pip install pyunpack
!pip install patool
from pyunpack.cli import Archive
os.system('apt-get install p7zip')
print(os.getcwd())
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/
Collecting lightfm
  Downloading lightfm-1.17.tar.gz (316 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 316.4/316.4 kB 8.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from light
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from li
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packa
Building wheels for collected packages: lightfm
  Building wheel for lightfm (setup.py) ... done
  Created wheel for lightfm: filename=lightfm-1.17-cp310-cp310-linux_x86_64.whl size=867220
  Stored in directory: /root/.cache/pip/wheels/4f/9b/7e/0b256f2168511d8fa4dae4fae0200fdbd72
Successfully built lightfm
Installing collected packages: lightfm
Successfully installed lightfm-1.17
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/
Collecting pyunpack
  Downloading pyunpack-0.3-py2.py3-none-any.whl (4.1 kB)
Collecting easyprocess (from pyunpack)
  Downloading EasyProcess-1.1-py3-none-any.whl (8.7 kB)
Collecting entrypoint2 (from pyunpack)
  Downloading entrypoint2-1.1-py2.py3-none-any.whl (9.9 kB)
Installing collected packages: entrypoint2, easyprocess, pyunpack
Successfully installed easyprocess-1.1 entrypoint2-1.1 pyunpack-0.3
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/
Collecting patool
  Downloading patool-1.12-py2.py3-none-any.whl (77 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 77.5/77.5 kB 3.6 MB/s eta 0:00:00
Installing collected packages: patool
Successfully installed patool-1.12
/content
```

```
train = pd.read_csv('/content/train.csv')
test = pd.read_csv('/content/test.csv')
songs = pd.read_csv('/content/songs.csv')
members = pd.read_csv('/content/members.csv')

print('Data loading completed!')
print(train.shape, test.shape, songs.shape, members.shape)
```

```
Data loading completed!
(23372, 6) (15573, 6) (31674, 7) (34403, 7)
```

```
print(train.columns)
print(test.columns)
print(songs.columns)
print(members.columns)
```

```
Index(['msno', 'song_id', 'source_system_tab', 'source_screen_name',
       'source_type', 'target'],
      dtype='object')
Index(['id', 'msno', 'song_id', 'source_system_tab', 'source_screen_name',
       'source_type'],
      dtype='object')
Index(['song_id', 'song_length', 'genre_ids', 'artist_name', 'composer',
       'lyricist', 'language'],
      dtype='object')
Index(['msno', 'city', 'bd', 'gender', 'registered_via',
       'registration_init_time', 'expiration_date'],
      dtype='object')
```

```
song_cols = ['song_id', 'song_length', 'genre_ids', 'artist_name', 'composer', 'language']
train = train.merge(songs[song_cols], on='song_id', how='left')
test = test.merge(songs[song_cols], on='song_id', how='left')

mem_cols = ['msno', 'city', 'bd', 'gender']
train = train.merge(members[mem_cols], on='msno', how='left')
test = test.merge(members[mem_cols], on='msno', how='left')

for col in [['msno', 'song_id', 'source_system_tab', 'source_screen_name',
             'source_type', 'genre_ids', 'artist_name',
             'composer', 'language', 'city', 'gender']]:
            train[col] = train[col].astype('category')
            test[col] = test[col].astype('category')
```

```
for col in train.columns:
    print(train[col].value_counts(), "\n")
```

```
train = train.drop(['bd', 'msno', 'song_length', 'source_system_tab'], axis = 1)
test = test.drop(['bd', 'msno', 'song_length', 'source_system_tab'], axis = 1)
```

```
31      595
36      562
34      530
18      511
32      478
19      378
33      320
37      272
39      270
38      228
40      182
41      159
46      103
44       95
43       95
47       90
17       87
54       75
52       53
50       46
45       45
42       44
51       40
55       31
48       30
57       29
16       27
49       21
53       16
60       16
59       15
111      14
3         9
73        7
58        6
67        5
65        5
66        5
131       5
14        4
64        3
56        2
Name: bd, dtype: int64

female    7382
male      6956
Name: gender, dtype: int64
```

```
train.columns
```

```
Index(['song_id', 'source_screen_name', 'source_type', 'target', 'genre_ids',
       'artist_name', 'composer', 'language', 'city', 'gender'],
      dtype='object')
```

```
test.columns
```

```
Index(['id', 'song_id', 'source_screen_name', 'source_type', 'genre_ids',
       'artist_name', 'composer', 'language', 'city', 'gender'],
      dtype='object')
```

```
df_col = [ 'song_id', 'source_screen_name',
       'source_type', 'genre_ids', 'artist_name', 'language', 'city', 'gender']
train = train.drop(['composer'], axis=1)
test = test.drop(['composer'], axis=1)
from sklearn.preprocessing import LabelEncoder

for i in range(len(df_col)):
    train[df_col[i]] = LabelEncoder().fit_transform(train[df_col[i]])

for i in range(len(df_col)):
    test[df_col[i]] = LabelEncoder().fit_transform(test[df_col[i]])


from sklearn.impute import SimpleImputer
my_imputer = SimpleImputer()
train = my_imputer.fit_transform(train)

my_imputer = SimpleImputer()
test = my_imputer.fit_transform(test)


train
```

```
array([[2.284e+03, 6.000e+00, 5.000e+00, ..., 8.000e+00, 0.000e+00,
        2.000e+00],
       [6.754e+03, 7.000e+00, 4.000e+00, ..., 8.000e+00, 1.100e+01,
        0.000e+00],
       [3.644e+03, 7.000e+00, 4.000e+00, ..., 8.000e+00, 1.100e+01,
        0.000e+00],
       ...,
       [5.231e+03, 1.000e+01, 5.000e+00, ..., 8.000e+00, 0.000e+00,
        2.000e+00],
       [3.483e+03, 3.000e+00, 8.000e+00, ..., 8.000e+00, 0.000e+00,
        2.000e+00],
       [4.626e+03, 1.700e+01, 1.100e+01, ..., 8.000e+00, 0.000e+00,
        2.000e+00]])
```

```
test
```

```
array([[0.0000e+00, 4.5910e+03, 7.0000e+00, ..., 8.0000e+00, 0.0000e+00,
        2.0000e+00],
       [1.0000e+00, 8.2810e+03, 7.0000e+00, ..., 1.0000e+00, 0.0000e+00,
        2.0000e+00],
       [2.0000e+00, 1.4120e+03, 1.7000e+01, ..., 3.0000e+00, 0.0000e+00,
        2.0000e+00],
       ...,
       [1.5570e+04, 1.3020e+03, 1.0000e+01, ..., 8.0000e+00, 1.1000e+01,
        0.0000e+00],
       [1.5571e+04, 5.9310e+03, 1.0000e+01, ..., 8.0000e+00, 1.1000e+01,
        0.0000e+00],
       [1.5572e+04, 8.5690e+03, 1.7000e+01, ..., 8.0000e+00, 2.1000e+01,
        2.0000e+00]])
```

```
train = pd.DataFrame(train, columns = [ 'song_id', 'source_screen_name','source_type',
                                       'target',  'genre_ids', 'artist_name', 'language',
                                       'city', 'gender'])
test = pd.DataFrame(test, columns = ['id', 'song_id', 'source_screen_name','source_type',
                                       'genre_ids', 'artist_name', 'language',
                                       'city', 'gender'])
```

```
test
```

	id	song_id	source_screen_name	source_type	genre_ids	artist_name	language
0	0.0	4591.0	7.0	3.0	54.0	550.0	8.0
1	1.0	8281.0	7.0	3.0	33.0	393.0	1.0
2	2.0	1412.0	17.0	9.0	14.0	274.0	3.0
3	3.0	8552.0	12.0	7.0	54.0	550.0	8.0
4	4.0	3259.0	12.0	7.0	54.0	550.0	8.0
...	...	...	...	...	...	...	...
15568	15568.0	1538.0	7.0	3.0	54.0	550.0	8.0
15569	15569.0	4007.0	10.0	9.0	54.0	550.0	8.0
15570	15570.0	1302.0	10.0	9.0	54.0	550.0	8.0
15571	15571.0	5931.0	10.0	9.0	54.0	550.0	8.0
15572	15572.0	8569.0	17.0	12.0	54.0	550.0	8.0

15573 rows × 9 columns

```
train = train.astype(int)
test = test.astype(int)
```

```
from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

X = train
X = X.drop(['target'], axis = 1)
y = train[['target']]

print(X.head())
print(y.head())
```

```
   song_id  source_screen_name  source_type  genre_ids  artist_name  language  \
0     2284                   6            5         59          693         8
1     6754                   7            4         59          693         8
2     3644                   7            4         59          693         8
3      719                   7            4         59          693         8
4     1043                   6            5         59          693         8

   city  gender
0     0       2
1    11       0
2    11       0
3    11       0
4     0       2
   target
0       1
1       1
2       1
3       1
4       1
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30)
clf = RandomForestClassifier(n_estimators = 16)
clf.fit(X_train, y_train.values.ravel())
y_pred = clf.predict(X_test)
from sklearn import metrics
print()

# using metrics module for accuracy calculation
print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))
```

```
ACCURACY OF THE MODEL:  0.7357387335995437
```

```
pred = clf.predict(test.drop(['id'], axis = 1))
```

```
subm = pd.DataFrame()
subm['id'] = test['id']
subm['target'] = pred

subm
```

	id	target
0	0	1
1	1	1
2	2	0
3	3	1
4	4	0
...	...	...
15568	15568	1
15569	15569	0
15570	15570	0
15571	15571	0
15572	15572	1

15573 rows × 2 columns

✓ 0s completed at 8:20 PM ● ×