hw2

July 8, 2021

[187]:

```
# !pip install imblearn
```

```
Collecting imblearn
  Downloading imblearn-0.0-py2.py3-none-any.whl (1.9 kB)
Collecting imbalanced-learn
  Downloading imbalanced_learn-0.8.0-py3-none-any.whl (206 kB)
Requirement already satisfied: joblib>=0.11 in
c:\users\jack\miniconda3\envs\datascience\lib\site-packages (from imbalanced-
learn->imblearn) (1.0.1)
Requirement already satisfied: scikit-learn>=0.24 in
c:\users\jack\miniconda3\envs\datascience\lib\site-packages (from imbalanced-
learn->imblearn) (0.24.2)
Requirement already satisfied: numpy>=1.13.3 in
c:\users\jack\miniconda3\envs\datascience\lib\site-packages (from imbalanced-
learn->imblearn) (1.19.2)
Requirement already satisfied: scipy>=0.19.1 in
c:\users\jack\miniconda3\envs\datascience\lib\site-packages (from imbalanced-
learn->imblearn) (1.6.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\users\jack\miniconda3\envs\datascience\lib\site-packages (from scikit-
learn>=0.24->imbalanced-learn->imblearn) (2.1.0)
Installing collected packages: imbalanced-learn, imblearn
Successfully installed imbalanced-learn-0.8.0 imblearn-0.0
```

[1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import lightgbm as lgb
from collections import Counter
from matplotlib import pyplot as plt
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split, cross_val_score,␣
 ↪GridSearchCV
from sklearn.metrics import roc_auc_score, accuracy_score

%matplotlib inline
```

```
[2]: train_data = pd.read_csv(r'train_final.csv')
     test_data = pd.read_csv(r'test_final.csv')
```

50000   146   loan_status

```
[3]: print("train data size: ", train_data.shape)
     print("test data size: ", test_data.shape)
```

```
train data size:  (50000, 146)
test data size:  (50000, 146)
```

```
[4]: train_data['loan_status'].head()
```

```
[4]: 0    1
     1    1
     2    1
     3    1
     4    1
     Name: loan_status, dtype: int64
```

```
[5]: train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Columns: 146 entries, continuous_annual_inc to discrete_term_2_one_hot
dtypes: float64(20), int64(126)
memory usage: 55.7 MB
```

```
[77]: # [idx for idx in train_data.columns if train_data[idx].dtype == np.float64]
```

0.1

```
[7]: train_data.isnull().sum().sort_values(ascending=False)[:10]
```

```
[7]: continuous_annual_inc_joint              49780
     continuous_dti_joint                     49780
     continuous_mths_since_last_record        40505
     continuous_mths_since_last_major_derog   34948
     continuous_mths_since_last_delinq        23917
     continuous_dti                               1
     discrete_addr_state_29_one_hot               0
     discrete_addr_state_34_one_hot               0
     discrete_addr_state_33_one_hot               0
     discrete_addr_state_32_one_hot               0
     dtype: int64
```

```
[8]: sum(train_data['continuous_annual_inc_joint'].isnull() &␣
      ↪train_data['loan_status'] == 1) /␣
      ↪sum(train_data['continuous_annual_inc_joint'].isnull()),\
sum(train_data['continuous_annual_inc_joint'].notnull() &␣
 ↪train_data['loan_status'] == 1) /␣
 ↪sum(train_data['continuous_annual_inc_joint'].notnull())
```

[8]: (0.7960024106066693, 0.740909090909091)

[9]:
```
sum(train_data['continuous_dti_joint'].isnull() & train_data['loan_status'] ==␣
 ↪1)/sum(train_data['continuous_dti_joint'].isnull()),\
sum(train_data['continuous_dti_joint'].notnull() & train_data['loan_status'] ==␣
 ↪1)/sum(train_data['continuous_dti_joint'].notnull())
```

[9]: (0.7960024106066693, 0.740909090909091)

[10]:
```
record = sum(train_data['continuous_mths_since_last_record'].isnull())
TP = sum(train_data['continuous_mths_since_last_record'].isnull() &␣
 ↪train_data['loan_status'] == 1)
precision = TP / sum(train_data['continuous_mths_since_last_record'].isnull())
recall = TP / sum(train_data['loan_status'] == 1)
2 * precision * recall / (precision + recall)
```

[10]: 0.8095848953208872

[11]:
```
sum(train_data['continuous_mths_since_last_record'].isnull() &␣
 ↪train_data['loan_status'] == 1) /␣
 ↪sum(train_data['continuous_mths_since_last_record'].isnull()),\
sum(train_data['continuous_mths_since_last_record'].notnull() &␣
 ↪train_data['loan_status'] == 1) /␣
 ↪sum(train_data['continuous_mths_since_last_record'].notnull())
```

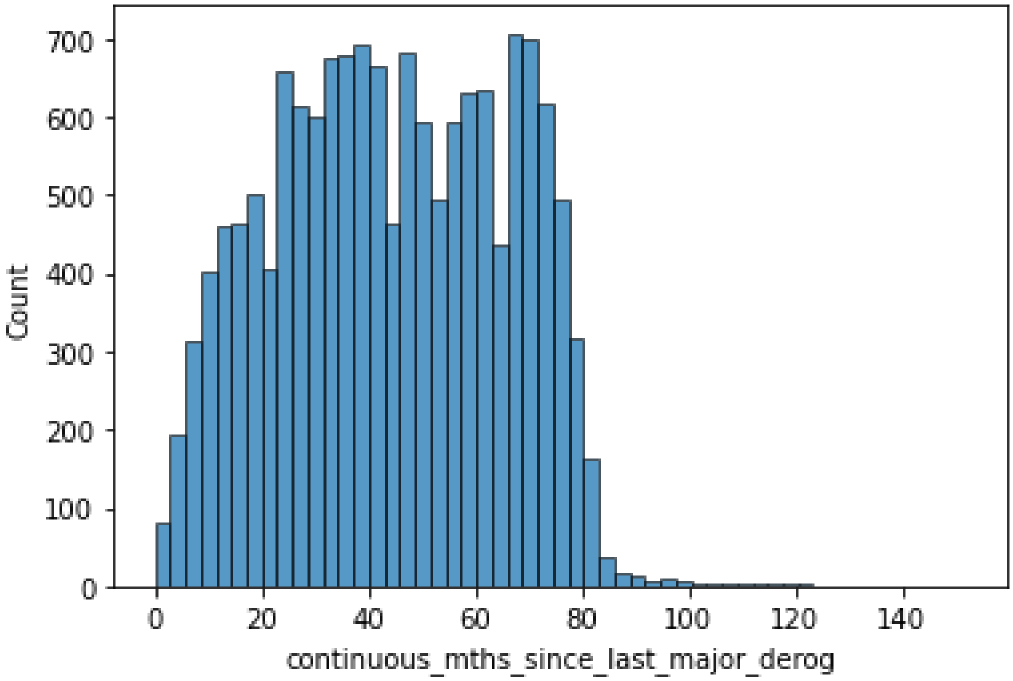
[11]: (0.8024194543883472, 0.7673512374934176)

[12]:
```
major_derog = sum(train_data['continuous_mths_since_last_major_derog'].isnull())
major_derog_loan = sum(train_data['continuous_mths_since_last_major_derog'].
 ↪isnull() & train_data['loan_status'] == 1)
major_derog, major_derog_loan, major_derog_loan / major_derog
```

[12]: (34948, 28055, 0.802764106672771)

[13]:
```
sns.histplot(train_data['continuous_mths_since_last_major_derog'])
```

[13]: <AxesSubplot:xlabel='continuous_mths_since_last_major_derog', ylabel='Count'>

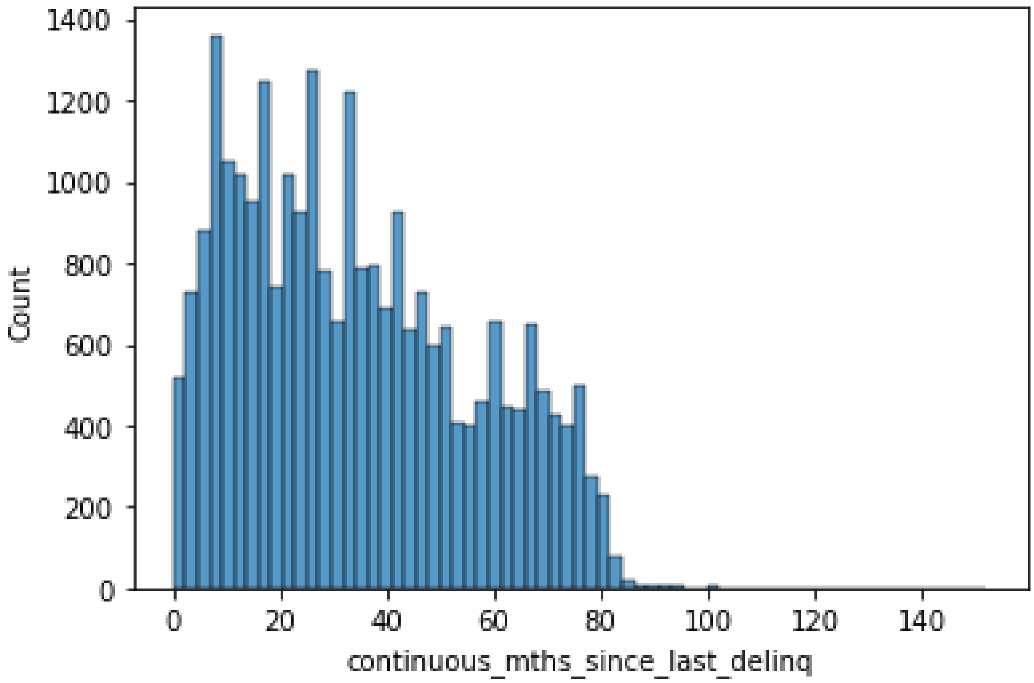


[14]:
```
delinq = sum(train_data['continuous_mths_since_last_delinq'].isnull())
delinq_loan = sum(train_data['continuous_mths_since_last_delinq'].isnull() &␣
 ↪train_data['loan_status'] == 1)
delinq, delinq_loan, delinq_loan / delinq
```

[14]:
```
(23917, 19200, 0.8027762679265794)
```

[15]:
```
sns.histplot(train_data['continuous_mths_since_last_delinq'])
```

[15]:
```
<AxesSubplot:xlabel='continuous_mths_since_last_delinq', ylabel='Count'>
```



0.2

- continuous_dti
- continuous_annual_inc_joint continuous_dti_joint null
- null binary

[16]:
```
train_data.fillna(value={'continuous_dti': train_data['continuous_dti'].
 ↪mean()}, inplace=True)
train_data['continuous_mths_since_last_record_isnull'] =␣
 ↪train_data['continuous_mths_since_last_record'].isnull()
train_data['continuous_mths_since_last_major_derog_isnull'] =␣
 ↪train_data['continuous_mths_since_last_major_derog'].isnull()
train_data['continuous_mths_since_last_delinq_isnull'] =␣
 ↪train_data['continuous_mths_since_last_delinq'].isnull()
train_data.drop(columns=['continuous_annual_inc_joint',
                         'continuous_dti_joint',
                         'continuous_mths_since_last_record',
                         'continuous_mths_since_last_major_derog',
                         'continuous_mths_since_last_delinq'], inplace=True)
```

[17]:
```
test_data.fillna(value={'continuous_dti': test_data['continuous_dti'].mean()},␣
 ↪inplace=True)
test_data['continuous_mths_since_last_record_isnull'] =␣
 ↪test_data['continuous_mths_since_last_record'].isnull()
```

```
test_data['continuous_mths_since_last_major_derog_isnull'] =␣
 ↪test_data['continuous_mths_since_last_major_derog'].isnull()
test_data['continuous_mths_since_last_delinq_isnull'] =␣
 ↪test_data['continuous_mths_since_last_delinq'].isnull()
test_data.drop(columns=['continuous_annual_inc_joint',
                        'continuous_dti_joint',
                        'continuous_mths_since_last_record',
                        'continuous_mths_since_last_major_derog',
                        'continuous_mths_since_last_delinq'], inplace=True)
```

[18]:
```
train_data.isnull().sum().sort_values(ascending=False)[:5]
```

[18]:
```
continuous_mths_since_last_delinq_isnull         0
continuous_mths_since_last_major_derog_isnull    0
discrete_addr_state_36_one_hot                   0
discrete_addr_state_35_one_hot                   0
discrete_addr_state_34_one_hot                   0
dtype: int64
```

[19]:
```
train_data.columns
```

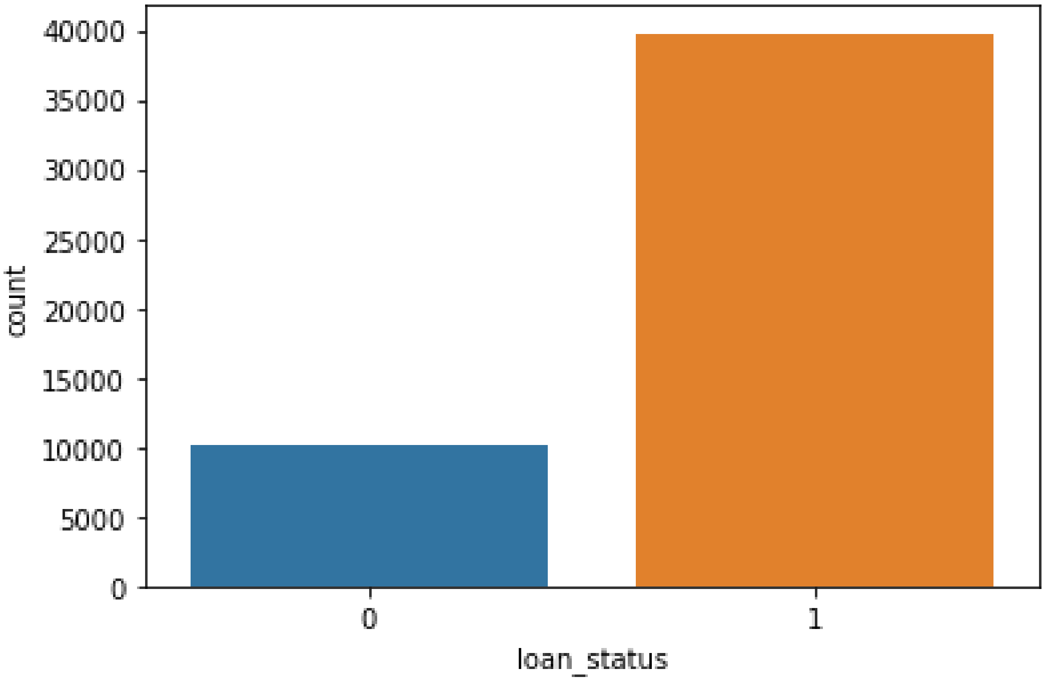
[19]:
```
Index(['continuous_annual_inc', 'continuous_delinq_2yrs', 'continuous_dti',
       'continuous_fico_range_high', 'continuous_fico_range_low',
       'continuous_funded_amnt', 'continuous_funded_amnt_inv',
       'continuous_inq_last_6mths', 'continuous_installment',
       'continuous_int_rate',
       …
       'discrete_sub_grade_31_one_hot', 'discrete_sub_grade_32_one_hot',
       'discrete_sub_grade_33_one_hot', 'discrete_sub_grade_34_one_hot',
       'discrete_sub_grade_35_one_hot', 'discrete_term_1_one_hot',
       'discrete_term_2_one_hot', 'continuous_mths_since_last_record_isnull',
       'continuous_mths_since_last_major_derog_isnull',
       'continuous_mths_since_last_delinq_isnull'],
      dtype='object', length=144)
```

0.3

loan_status 0 10000 1 40000 (SMOTE)

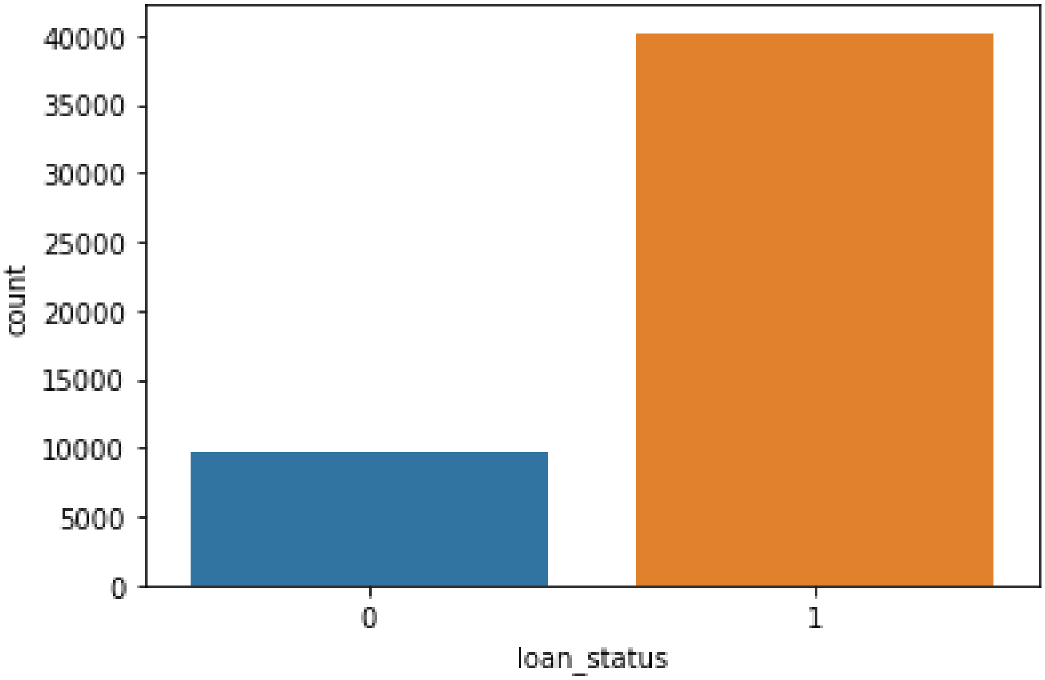
[20]:
```
sns.countplot(x='loan_status', data=train_data)
```

[20]:
```
<AxesSubplot:xlabel='loan_status', ylabel='count'>
```



```
[21]: sns.countplot(x='loan_status', data=test_data)
```

```
[21]: <AxesSubplot:xlabel='loan_status', ylabel='count'>
```



0.4 one hot

lightgbm

`LightGBM can use categorical features as input directly. It doesn't need to convert to one-hot`

[22]:
```
# one_hots = [idx for idx in train_data.columns if 'one_hot' in idx]
# one_hot_dict = {}
# for string in one_hots:
#     string = string[:-8]
#     n = int(string.split('_')[-1])
#     s = '_'.join(string.split('_')[:-1])
#     if s not in one_hot_dict.keys():
#         one_hot_dict[s] = n
#     else:
#         one_hot_dict[s] = max(one_hot_dict[s], n)
# one_hot_dict
```

[22]:
```
{'discrete_addr_state': 49,
 'discrete_application_type': 2,
 'discrete_emp_length': 12,
 'discrete_grade': 7,
 'discrete_home_ownership': 4,
 'discrete_policy_code': 1,
 'discrete_purpose': 12,
 'discrete_pymnt_plan': 1,
 'discrete_sub_grade': 35,
 'discrete_term': 2}
```

[105]:
```
# def helperOneHot(train_data, s, n):
#     for i in range(train_data.shape[0]):
#         for k in range(1, n + 1):
#             if train_data.loc[i, '{}_{}_one_hot'.format(s, k)] == 1:
#                 train_data.loc[i, s] = int(k)
#                 break
#     return train_data.astype({s: 'int64'})

# train_data_no_oh = train_data
# test_data_no_oh = test_data
# for s, n in one_hot_dict.items():
#     train_data_no_oh = helperOneHot(train_data_no_oh, s, n)
#     test_data_no_oh = helperOneHot(test_data_no_oh, s, n)

# for s in one_hot_dict.keys():
```

```
#     cols = ['{}_{}_one_hot'.format(s, i) for i in range(1, one_hot_dict[s] +␣
↪1)]
#     train_data_no_oh.drop(columns=cols, inplace=True)
#     test_data_no_oh.drop(columns=cols, inplace=True)
```

0.5

[106]:

```
# X_train_no_oh, y_train_no_oh = train_data_no_oh.drop(columns='loan_status'),␣
↪train_data_no_oh['loan_status']
# print(X_train_no_oh.shape, y_train_no_oh.shape)

# X_test_no_oh, y_test_no_oh = test_data_no_oh.drop(columns='loan_status'),␣
↪test_data_no_oh['loan_status']
# print(X_test_no_oh.shape, y_test_no_oh.shape)
```

```
(50000, 28) (50000,)
(50000, 28) (50000,)
```

[107]:

```
# counter = Counter(y_train)
# print(counter)
# # # transform the dataset
# oversample = SMOTE()
# X_train_smote, y_train_smote = oversample.fit_resample(X_train_no_oh,␣
↪y_train_no_oh)
# counter = Counter(y_train_smote)
# print(counter)
```

```
Counter({0: 20387, 1: 20358})
Counter({1: 39788, 0: 39788})
```

lgb.cv   GridSearchCV

[78]:

```
# X_train, X_val, y_train, y_val = train_test_split(X_train_no_oh,␣
↪y_train_no_oh, test_size=0.1, stratify=y_train_no_oh)
# X_train, X_val, y_train, y_val = train_test_split(X_train_no_oh,␣
↪y_train_no_oh, test_size=0.1)
# X_train, X_val, y_train, y_val = train_test_split(train_data.
↪drop(columns='loan_status'), train_data['loan_status'], test_size=0.1)
# X_train, X_val, y_train, y_val = train_test_split(X_train_smote,␣
↪y_train_smote, test_size=0.2, stratify=y_train_smote)

# print(X_train.head())
print(X_train.shape, X_val.shape)
```

```
(45000, 143) (5000, 143)
```

lgb.Dataset

[53]:

```
#   lgb.Dataset
# train_lgb = lgb.Dataset(X_train, label=y_train,␣
 ↪categorical_feature=list(one_hot_dict.keys()), free_raw_data=False)
# validation_lgb = lgb.Dataset(X_val, label=y_val,␣
 ↪categorical_feature=list(one_hot_dict.keys()), free_raw_data=False)
data_lgb = lgb.Dataset(train_data.drop(columns='loan_status'),␣
 ↪label=train_data['loan_status'], free_raw_data=False)
train_lgb = lgb.Dataset(X_train, label=y_train, free_raw_data=False)
validation_lgb = lgb.Dataset(X_val, label=y_val, free_raw_data=False)

#
params = {
    'boosting_type': 'gbdt',
    'objective': 'binary',
    'learning_rate': 0.1,
    'metric': 'auc',
    'num_iterations': 1000,
    'num_leaves': 20,
    'max_depth': 4,
    'subsample': 0.8,
    'colsample_bytree': 0.8
}
```

lgb.cv   n_estimators   83

[54]:

```
#
# gbm = lgb.train(params, train_lgb, valid_sets=[validation_lgb])


cv_results = lgb.cv(params, data_lgb, num_boost_round=1000, nfold=5,␣
 ↪stratified=False, shuffle=True, metrics='auc', early_stopping_rounds=50,␣
 ↪seed=0)
print('best n_estimators:', len(cv_results['auc-mean']))
print('best cv score:', pd.Series(cv_results['auc-mean']).max())
```

```
C:\Users\Jack\miniconda3\envs\datascience\lib\site-
packages\lightgbm\engine.py:527: UserWarning: Found `num_iterations` in params.
Will use it instead of argument
  _log_warning("Found `{}` in params. Will use it instead of
argument".format(alias))

[LightGBM] [Info] Number of positive: 31856, number of negative: 8144
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000933 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2152
[LightGBM] [Info] Number of data points in the train set: 40000, number of used
```

```
features: 139
[LightGBM] [Info] Number of positive: 31852, number of negative: 8148
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000876 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2152
[LightGBM] [Info] Number of data points in the train set: 40000, number of used
features: 139
[LightGBM] [Info] Number of positive: 31795, number of negative: 8205
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000933 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2152
[LightGBM] [Info] Number of data points in the train set: 40000, number of used
features: 139
[LightGBM] [Info] Number of positive: 31798, number of negative: 8202
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000831 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2152
[LightGBM] [Info] Number of data points in the train set: 40000, number of used
features: 139
[LightGBM] [Info] Number of positive: 31851, number of negative: 8149
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000895 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2152
[LightGBM] [Info] Number of data points in the train set: 40000, number of used
features: 139
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.796400 -> initscore=1.363944
[LightGBM] [Info] Start training from score 1.363944
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.796300 -> initscore=1.363328
[LightGBM] [Info] Start training from score 1.363328
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.794875 -> initscore=1.354565
[LightGBM] [Info] Start training from score 1.354565
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.794950 -> initscore=1.355025
[LightGBM] [Info] Start training from score 1.355025
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.796275 -> initscore=1.363174
[LightGBM] [Info] Start training from score 1.363174
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
best n_estimators: 83
best cv score: 0.9608433011135118
```

best `n_estimators` value is 83

`n_estimators` `max_depth` `num_leaves` 4 10

[56]:
```
paramTest1 = {'max_depth': range(2, 10, 1), 'num_leaves': range(5, 100, 5)}
```

```
gsearch1 = GridSearchCV(estimator = lgb.LGBMClassifier(boosting_type='gbdt',
                                                       objective='binary',
                                                       metrics='auc',
                                                       learning_rate=0.1,
                                                       n_estimators=83,
                                                       bagging_fraction = 0.8,
                                                       feature_fraction = 0.8),
                        param_grid=paramTest1, scoring='roc_auc', cv=5,␣
 ↪n_jobs=-1)
gsearch1.fit(train_data.drop(columns='loan_status'), train_data['loan_status'])
gsearch1.cv_results_['mean_test_score'], gsearch1.
 ↪cv_results_['mean_test_score'].mean()
```

```
[56]: (array([0.96068497, 0.96068497, 0.96068497, 0.96068497, 0.96068497,
                0.96068497, 0.96068497, 0.96068497, 0.96068497, 0.96068497,
                0.96068497, 0.96068497, 0.96068497, 0.96068497, 0.96068497,
                0.96068497, 0.96068497, 0.96068497, 0.96068497, 0.96082095,
                0.96103586, 0.96103586, 0.96103586, 0.96103586, 0.96103586,
                0.96103586, 0.96103586, 0.96103586, 0.96103586, 0.96103586,
                0.96103586, 0.96103586, 0.96103586, 0.96103586, 0.96103586,
                0.96103586, 0.96103586, 0.96103586, 0.96089721, 0.96111723,
                0.96088435, 0.96092834, 0.96092834, 0.96092834, 0.96092834,
                0.96092834, 0.96092834, 0.96092834, 0.96092834, 0.96092834,
                0.96092834, 0.96092834, 0.96092834, 0.96092834, 0.96092834,
                0.96092834, 0.96092834, 0.96089721, 0.96094652, 0.9609954 ,
                0.96082558, 0.96060316, 0.96056288, 0.96044602, 0.96044602,
                0.96044602, 0.96044602, 0.96044602, 0.96044602, 0.96044602,
                0.96044602, 0.96044602, 0.96044602, 0.96044602, 0.96044602,
                0.96044602, 0.96089721, 0.96091934, 0.96084894, 0.96068191,
                0.96044873, 0.96032326, 0.96028322, 0.96019266, 0.95997514,
                0.96004   , 0.95999664, 0.95997875, 0.95995706, 0.95995706,
                0.95995706, 0.95995706, 0.95995706, 0.95995706, 0.95995706,
                0.96089721, 0.96097118, 0.96082643, 0.96061995, 0.96049768,
                0.96049996, 0.96001299, 0.95983117, 0.95990553, 0.95978005,
                0.95963691, 0.95962657, 0.95961114, 0.95944057, 0.95943203,
                0.95942918, 0.95953964, 0.95914864, 0.95947342, 0.96089721,
                0.96092858, 0.96093808, 0.96076017, 0.96044615, 0.96034374,
                0.96010861, 0.96000475, 0.95997212, 0.95987829, 0.95964574,
                0.95960674, 0.95937185, 0.95923325, 0.95915181, 0.95925201,
                0.95912804, 0.95930986, 0.95922137, 0.96089721, 0.96094146,
                0.96081751, 0.96083461, 0.96042486, 0.96040946, 0.96038425,
                0.96001428, 0.95980696, 0.95973247, 0.95944011, 0.95941865,
                0.95910425, 0.95961542, 0.95932939, 0.95928848, 0.95907419,
                0.9586812 , 0.95898002]),
       0.9603928990699145)
```

[57]:
```
clf1 = gsearch1.best_estimator_
clf1
```

[57]:
```
LGBMClassifier(bagging_fraction=0.8, feature_fraction=0.8, max_depth=4,
               metrics='auc', n_estimators=83, num_leaves=10,
               objective='binary')
```

optimal value for `max_depth` is 4 and optimal value for `num_leaves` is 10.

accuracy	0.91764	roc_auc	0.875699

[58]:
```
def results(clf, data):
    y_pred = clf.predict(data.drop(columns='loan_status'))
    y_pred = np.array(list(map(lambda x: 1 if x >= 0.5 else 0, y_pred)))
    y = data['loan_status']
    return {'accuracy': accuracy_score(y, y_pred),
            'roc_auc': roc_auc_score(y, y_pred)}

results(clf1, test_data)
```

[58]:
```
{'accuracy': 0.91764, 'roc_auc': 0.8756988382937406}
```

min_data_in_leaf	140

[59]:
```
paramTest2 = {'min_data_in_leaf': range(5, 155, 5)}

gsearch2 = GridSearchCV(estimator = lgb.LGBMClassifier(boosting_type='gbdt',
                                                       objective='binary',
                                                       metrics='auc',
                                                       learning_rate=0.1,
                                                       n_estimators=82,
                                                       max_depth=4,
                                                       num_leaves=10,
                                                       bagging_fraction = 0.8,
                                                       feature_fraction = 0.8),
                        param_grid=paramTest2, scoring='roc_auc', cv=5,␣
 ↪n_jobs=-1)
gsearch2.fit(X_train,y_train)
gsearch2.cv_results_['mean_test_score'], gsearch2.
 ↪cv_results_['mean_test_score'].mean()
```

```
[LightGBM] [Warning] feature_fraction is set=0.8, colsample_bytree=1.0 will be
ignored. Current value: feature_fraction=0.8
[LightGBM] [Warning] min_data_in_leaf is set=140, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=140
[LightGBM] [Warning] bagging_fraction is set=0.8, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.8
```

```
[59]: (array([0.96059155, 0.96057467, 0.96053096, 0.96057324, 0.96055448,
              0.96055347, 0.96061442, 0.96056771, 0.9606309 , 0.96072561,
              0.96068197, 0.96066348, 0.96071941, 0.9607305 , 0.96061674,
              0.96058627, 0.96060219, 0.96053595, 0.96059588, 0.96066609,
              0.9606667 , 0.96071893, 0.96077546, 0.96059836, 0.96064973,
              0.96074422, 0.96067689, 0.96078756, 0.96063708, 0.96052441]),
       0.9606364933221986)
```

```
[60]: gsearch2.best_params_
```

```
[60]: {'min_data_in_leaf': 140}
```

```
[61]: clf2 = gsearch2.best_estimator_
      clf2
```

```
[61]: LGBMClassifier(bagging_fraction=0.8, feature_fraction=0.8, max_depth=4,
                     metrics='auc', min_data_in_leaf=140, n_estimators=82,
                     num_leaves=10, objective='binary')
```

```
[62]: results(clf2, test_data)
```

```
[62]: {'accuracy': 0.91762, 'roc_auc': 0.8753765976682328}
```

lambda_l1   lambda_l2

```
[63]: paramTest3 = {'lambda_l1': [1e-5, 1e-3, 1e-1, 0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.
       ↪0],
                    'lambda_l2': [1e-5, 1e-3, 1e-1, 0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0]
      }

      gsearch3 = GridSearchCV(estimator = lgb.LGBMClassifier(boosting_type='gbdt',
                                                             objective='binary',
                                                             metrics='auc',
                                                             learning_rate=0.1,
                                                             n_estimators=82,
                                                             min_data_in_leaf=140,
                                                             max_depth=4,
                                                             num_leaves=10,
                                                             bagging_fraction = 0.8,
                                                             feature_fraction = 0.8),
                              param_grid=paramTest3, scoring='roc_auc', cv=5,␣
       ↪n_jobs=-1)
      gsearch3.fit(X_train,y_train)
      gsearch3.cv_results_['mean_test_score'], gsearch3.
       ↪cv_results_['mean_test_score'].mean()
```

```
[LightGBM] [Warning] feature_fraction is set=0.8, colsample_bytree=1.0 will be
```

```
ignored. Current value: feature_fraction=0.8
[LightGBM] [Warning] min_data_in_leaf is set=140, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=140
[LightGBM] [Warning] lambda_l1 is set=0.001, reg_alpha=0.0 will be ignored.
Current value: lambda_l1=0.001
[LightGBM] [Warning] bagging_fraction is set=0.8, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.8
[LightGBM] [Warning] lambda_l2 is set=0.001, reg_lambda=0.0 will be ignored.
Current value: lambda_l2=0.001
```

[63]:

```
(array([0.96078755, 0.96078753, 0.96062171, 0.96078756, 0.96062171,
        0.96062813, 0.96063375, 0.9606609 , 0.96061072, 0.96056447,
        0.96077298, 0.96080488, 0.96060907, 0.96077296, 0.96060907,
        0.96062493, 0.96063252, 0.96067345, 0.96059741, 0.9605999 ,
        0.96063565, 0.96061885, 0.96070876, 0.96063565, 0.96070876,
        0.96069257, 0.96062434, 0.96061176, 0.96060351, 0.96067026,
        0.96078756, 0.96078753, 0.96062171, 0.96078756, 0.96062171,
        0.96062813, 0.96063374, 0.96066087, 0.96061072, 0.96056447,
        0.96063565, 0.96061885, 0.96070876, 0.96063565, 0.96070876,
        0.96069257, 0.96062434, 0.96061176, 0.96060351, 0.96067026,
        0.96065666, 0.96065662, 0.96069431, 0.96065666, 0.96069431,
        0.96055566, 0.96072729, 0.96076221, 0.96072778, 0.9606771 ,
        0.96061636, 0.96061637, 0.96061122, 0.96061634, 0.96061122,
        0.96058555, 0.96070384, 0.96062638, 0.96061826, 0.96062976,
        0.96076249, 0.96076236, 0.96075459, 0.96076249, 0.96075459,
        0.96063745, 0.96065886, 0.96065782, 0.96059195, 0.96060302,
        0.96062758, 0.96062766, 0.96067657, 0.96062758, 0.96067657,
        0.96060148, 0.9606397 , 0.96058609, 0.96073001, 0.96075929,
        0.96069248, 0.96069239, 0.96061304, 0.96069248, 0.96061304,
        0.96064063, 0.96062726, 0.96068497, 0.96068147, 0.96067488]),
 0.9606628013226058)
```

[64]:

```
gsearch3.best_params_
```

[64]:

```
{'lambda_l1': 0.001, 'lambda_l2': 0.001}
```

[65]:

```
clf3 = gsearch3.best_estimator_
clf3
```

[65]:

```
LGBMClassifier(bagging_fraction=0.8, feature_fraction=0.8, lambda_l1=0.001,
               lambda_l2=0.001, max_depth=4, metrics='auc',
               min_data_in_leaf=140, n_estimators=82, num_leaves=10,
               objective='binary')
```

[66]:

```
results(clf3, test_data)
```

[66]:

```
{'accuracy': 0.91762, 'roc_auc': 0.8753765976682328}
```

`learning_rate` 0.01 `n_estimators` 800

[76]:

```
clf = lgb.LGBMClassifier(
    learning_rate=0.01,
    n_estimators=800,
    bagging_fraction=0.8,
    feature_fraction=0.8,
#     lambda_l1=0.1,
#     lambda_l2=0.5,
    max_depth=4,
    boosting_type='gbdt',
    metrics='auc',
    min_data_in_leaf=140,
    num_leaves=10,
    objective='binary'
)
clf.fit(X_train, y_train)
results(clf, test_data)
```

```
[LightGBM] [Warning] feature_fraction is set=0.8, colsample_bytree=1.0 will be
ignored. Current value: feature_fraction=0.8
[LightGBM] [Warning] min_data_in_leaf is set=140, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=140
[LightGBM] [Warning] bagging_fraction is set=0.8, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.8
```

[76]:

```
{'accuracy': 0.91788, 'roc_auc': 0.8753445529179209}
```

accuray 0.91788 roc_auc 0.875345

[ ]: