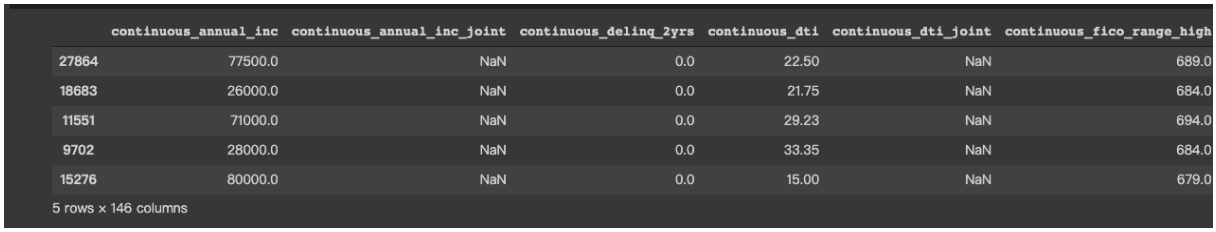
毕业设计

目标：根据 Loan Club 数据集构建衍生变量并运用 stacking 提升效果。

导入数据集后，首先对train训练集做一次shuffle，使数据分布随机且平均。

```
train=train.sample(frac=1)
```



	continuous_annual_inc	continuous_annual_inc_joint	continuous_delinq_2yrs	continuous_dti	continuous_dti_joint	continuous_fico_range_high
27864	77500.0	NaN	0.0	22.50	NaN	689.0
18683	26000.0	NaN	0.0	21.75	NaN	684.0
11551	71000.0	NaN	0.0	29.23	NaN	694.0
9702	28000.0	NaN	0.0	33.35	NaN	684.0
15276	80000.0	NaN	0.0	15.00	NaN	679.0

5 rows × 146 columns

baseline的模型训练

参数设置

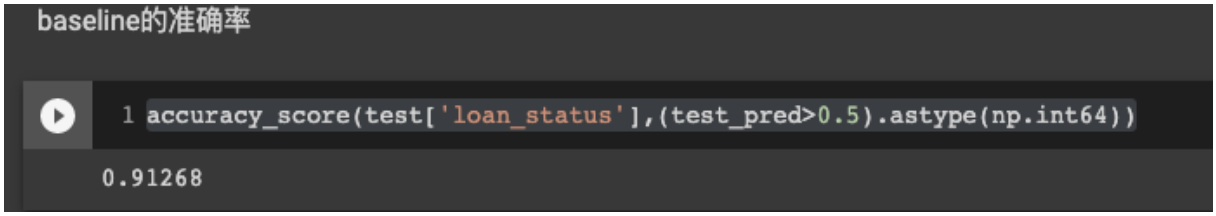
```
from sklearn.model_selection import KFold
kfold = KFold(n_splits=5)
params = {'num_thread': 4, 'num_leaves': 64, 'metric':
'quantile', 'objective': 'binary',
          'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
print('参数设置: ')
print(params)
train_pred, test_pred, error_rate, models =
fitter.train_k_fold(kfold, train, test, params = params)
print('mean of error_rate : ')
print(np.mean(error_rate))
```

得到baseline的mean of error_rate

```
mean of error_rate :
0.08657999999999999
```

得到baseline的accuracy score = 0.91268

```
accuracy_score(test['loan_status'],
(test_pred>0.5).astype(np.int64))
```

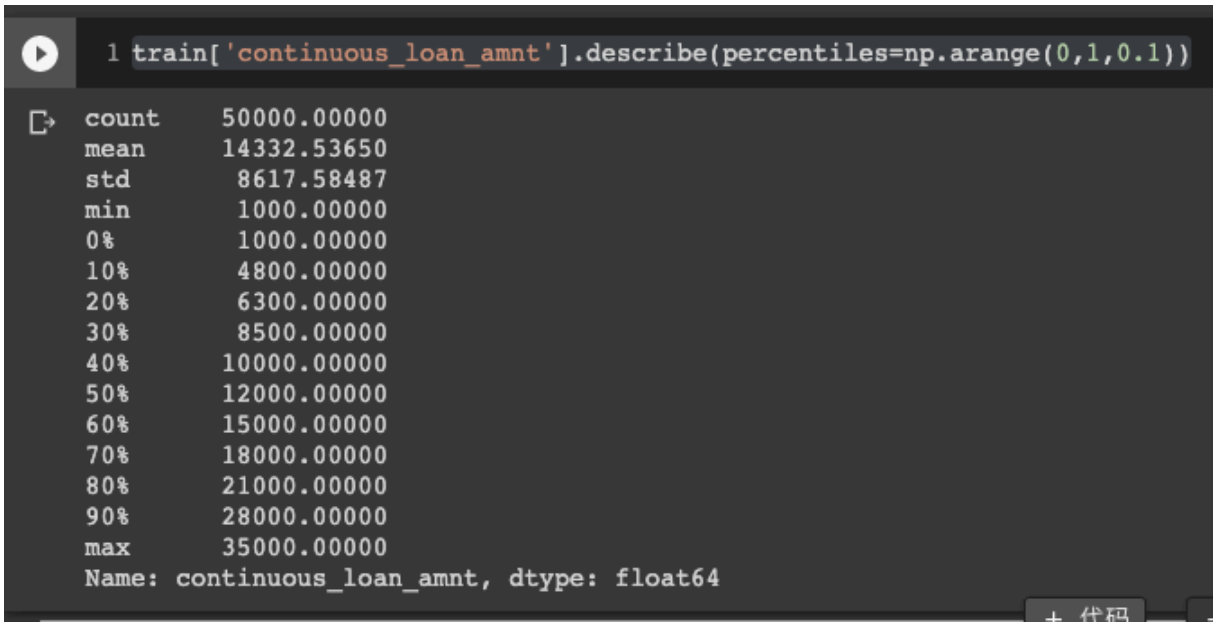


构建衍生变量 is_loan_amnt_10000

对continuous_loan_amnt进行数据探索

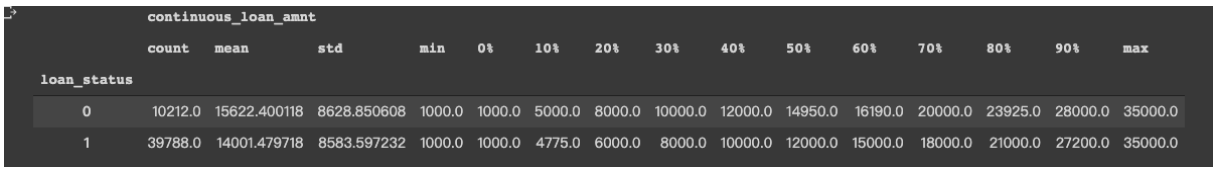
通过describe方法观测其分布

```
train['continuous_loan_amnt'].describe(percentiles=np.arange(0,1,
0.1))
```



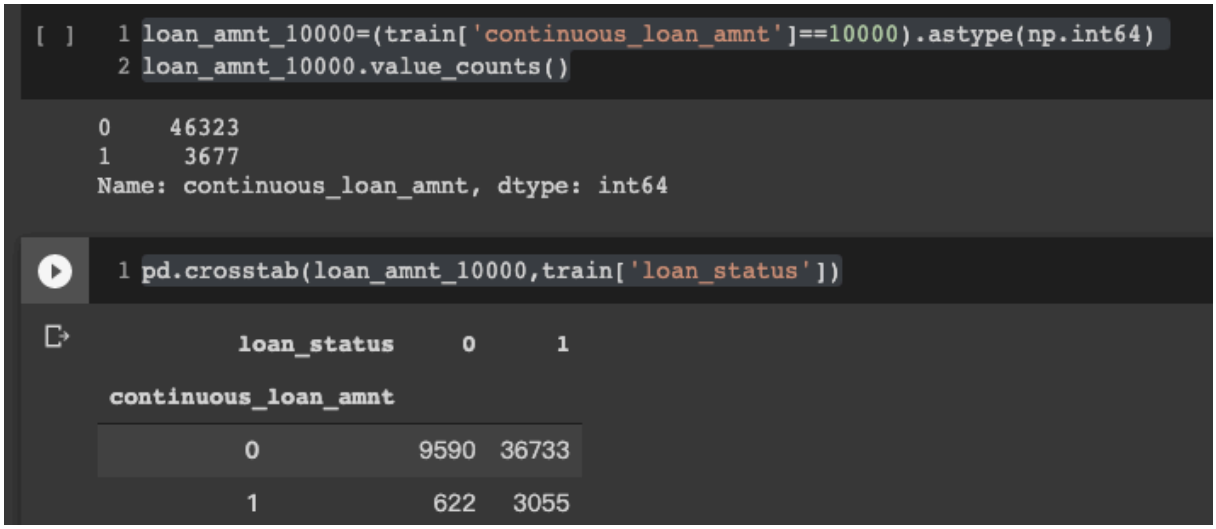
观测continuous_loan_amnt与loan_status之间的关系

```
train[['loan_status','continuous_loan_amnt']].groupby('loan_status').describe(percentiles=np.arange(0,1,0.1))
```



	continuous_loan_amnt														
	count	mean	std	min	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	max
loan_status															
0	10212.0	15622.400118	8628.850608	1000.0	1000.0	5000.0	8000.0	10000.0	12000.0	14950.0	16190.0	20000.0	23925.0	28000.0	35000.0
1	39788.0	14001.479718	8583.597232	1000.0	1000.0	4775.0	6000.0	8000.0	10000.0	12000.0	15000.0	18000.0	21000.0	27200.0	35000.0

```
loan_amnt_10000=(train['continuous_loan_amnt']==10000).astype(np.int64)
loan_amnt_10000.value_counts()
pd.crosstab(loan_amnt_10000,train['loan_status'])
```



```
[ ]  1 loan_amnt_10000=(train['continuous_loan_amnt']==10000).astype(np.int64)
     2 loan_amnt_10000.value_counts()
```

```
0    46323
1     3677
Name: continuous_loan_amnt, dtype: int64
```

```
1 pd.crosstab(loan_amnt_10000,train['loan_status'])
```

loan_status	0	1
continuous_loan_amnt		
0	9590	36733
1	622	3055

生成新的训练集train1和测试集test1

```
data1=pd.concat([train,test],axis=0)
data1.shape
data1['is_loan_amnt_10000']=(data1['continuous_loan_amnt']==10000).astype(np.int64)
train1=data1.iloc[:TRAIN_IDX, :]
test1 = data1.iloc[TRAIN_IDX:TEST_IDX, :]
```

新数据集进行调参，并进行模型训练

通过对numleaves进行调参，观察accuracy score；其他参数继承baseline的参数设置。

参数设置

```
from sklearn.model_selection import KFold
kfold = KFold(n_splits=5)
for num_leaves in [12, 31, 62, 81, 127]:
  params = {'num_thread': 4, 'num_leaves': num_leaves, 'metric':
'quantile', 'objective': 'binary',
            'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
  print('参数设置: ')
  print(params)
  train_pred, test_pred, error_rate, models =
fitter.train_k_fold(kfold, train1, test1, params = params)
  print('mean of error_rate : ')
  print(np.mean(error_rate))
  print('accuracy score: ')
  print(accuracy_score(test1['loan_status'],
(test_pred>0.5).astype(np.int64)))
```

模型训练结果，当num_leaves=12时，accuracy scroe=0.91362 为最好结果

```
参数设置:
{'num_thread': 4, 'num_leaves': 12, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08732
accuracy score:
0.91362
参数设置:
{'num_thread': 4, 'num_leaves': 31, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08718
accuracy score:
0.91264
参数设置:
{'num_thread': 4, 'num_leaves': 62, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08664000000000001
accuracy score:
0.9128
参数设置:
```

```
{'num_thread': 4, 'num_leaves': 81, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08680000000000003
accuracy score:
0.91276
参数设置:
{'num_thread': 4, 'num_leaves': 127, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.0867
accuracy score:
0.91218
```

构建另一个衍生变量 is_funded_amnt_inv_10000

通过构建另一个新的衍生变量，观察能否对模型训练结果提升准确率。

对continuous_funded_amnt_inv进行数据探索

```
train1['continuous_funded_amnt_inv'].describe(percentiles=np.aran
ge(0,1,0.05))
```

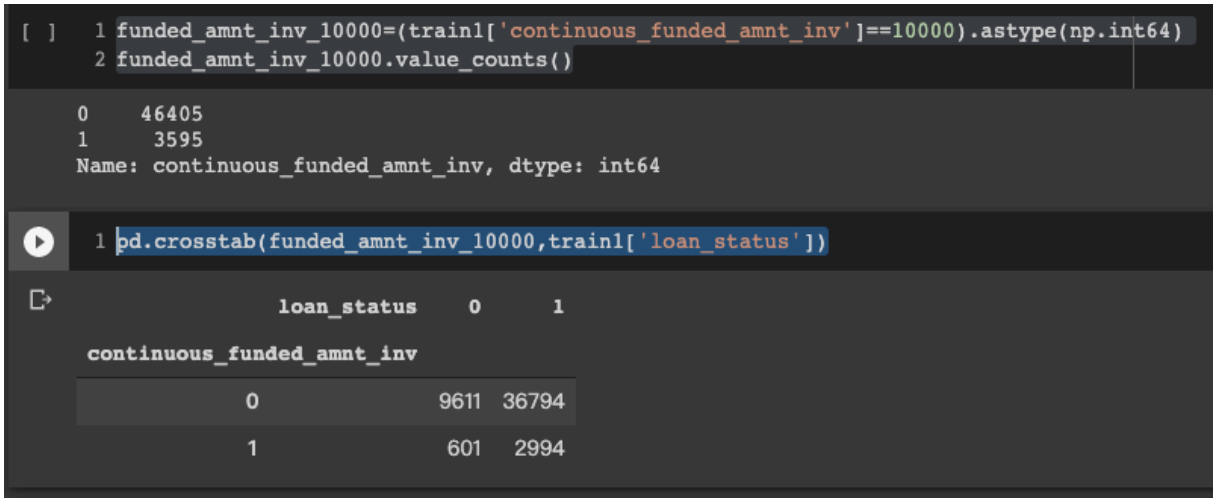
```
count    50000.000000
mean     14325.533000
std       8612.853833
min        950.000000
0%         950.000000
5%        3000.000000
10%       4800.000000
15%       5500.000000
20%       6300.000000
25%       7750.000000
30%       8500.000000
35%      10000.000000
40%      10000.000000
45%      11600.000000
50%      12000.000000
55%      14000.000000
60%      15000.000000
65%      16000.000000
70%      18000.000000
75%      20000.000000
80%      21000.000000
85%      24000.000000
90%      27900.000000
95%      32000.000000
max      35000.000000
Name: continuous_funded_amnt_inv, dtype: float64
```

```
train1[['loan_status','continuous_funded_amnt_inv']].groupby('loan_status').describe(percentiles=np.arange(0,1,0.05))
```

	continuous_funded_amnt_inv																
	count	mean	std	min	0%	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%
loan_status																	
0	10212.0	15610.490110	8622.260795	950.0	950.0	3675.0	5000.0	6025.0	8000.0	9150.0	10000.0	11000.0	12000.0	13100.0	14850.0	15000.0	16150.
1	39788.0	13995.735523	8579.565892	1000.0	1000.0	3000.0	4750.0	5000.0	6000.0	7200.0	8000.0	9600.0	10000.0	10850.0	12000.0	13200.0	15000.

```
funded_amnt_inv_10000=(train1['continuous_funded_amnt_inv']==10000).astype(np.int64)
```

```
funded_amnt_inv_10000.value_counts()
pd.crosstab(funded_amnt_inv_10000,train1['loan_status'])
```



生成新的训练集train2和测试集test2

通过对num_leaves、metric参数进行调参，训练train2并观测accuracy score；其他参数继承上一轮参数的设置。

参数设置

```
from sklearn.model_selection import KFold
kfold = KFold(n_splits=5)
for num_leaves in [12, 31, 62, 81]:
  for metric in ['l1', 'rmse', 'quantile', 'mape', 'binary']:
    params = {'num_thread': 4, 'num_leaves': num_leaves,
'metric': metric, 'objective': 'binary',
              'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
    print('参数设置: ')
    print(params)
    train_pred, test_pred, error_rate, models =
fitter.train_k_fold(kfold, train2, test2, params = params)
    print('mean of error_rate : ')
    print(np.mean(error_rate))
    print('accuracy score: ')
    print(accuracy_score(test2['loan_status'],
(test_pred>0.5).astype(np.int64)))
```

模型训练结果

从结果看，新衍生变量在整体降低了error rate，提升了accuracy score，但不同的metric参数的结果都一致。当num_leaves=12时，accuracy score=0.91368为最好的训练结果。

```
参数设置：
{'num_thread': 4, 'num_leaves': 12, 'metric': 'l1', 'objective':
'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.0869
accuracy score:
0.91368
参数设置：
{'num_thread': 4, 'num_leaves': 12, 'metric': 'rmse',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.0869
accuracy score:
0.91368
参数设置：
{'num_thread': 4, 'num_leaves': 12, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
```

```
mean of error_rate :
0.0869
accuracy score:
0.91368
参数设置:
{'num_thread': 4, 'num_leaves': 12, 'metric': 'mape',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.0869
accuracy score:
0.91368
参数设置:
{'num_thread': 4, 'num_leaves': 12, 'metric': 'binary',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.0869
accuracy score:
0.91368
参数设置:
{'num_thread': 4, 'num_leaves': 31, 'metric': 'l1', 'objective':
'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08686
accuracy score:
```

```
0.91306
参数设置：
{'num_thread': 4, 'num_leaves': 31, 'metric': 'rmse',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08686
accuracy score:
0.91306
参数设置：
{'num_thread': 4, 'num_leaves': 31, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08686
accuracy score:
0.91306
参数设置：
{'num_thread': 4, 'num_leaves': 31, 'metric': 'mape',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08686
accuracy score:
0.91306
参数设置：
```

```
{'num_thread': 4, 'num_leaves': 31, 'metric': 'binary',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08686
accuracy score:
0.91306
参数设置:
{'num_thread': 4, 'num_leaves': 62, 'metric': 'l1', 'objective':
'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08642000000000001
accuracy score:
0.91306
参数设置:
{'num_thread': 4, 'num_leaves': 62, 'metric': 'rmse',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08642000000000001
accuracy score:
0.91306
参数设置:
{'num_thread': 4, 'num_leaves': 62, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
```

```
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08642000000000001
accuracy score:
0.91306
参数设置:
{'num_thread': 4, 'num_leaves': 62, 'metric': 'mape',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08642000000000001
accuracy score:
0.91306
参数设置:
{'num_thread': 4, 'num_leaves': 62, 'metric': 'binary',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08642000000000001
accuracy score:
0.91306
参数设置:
{'num_thread': 4, 'num_leaves': 81, 'metric': 'l1', 'objective':
'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
```

```
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08634000000000001
accuracy score:
0.91324
参数设置:
{'num_thread': 4, 'num_leaves': 81, 'metric': 'rmse',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08634000000000001
accuracy score:
0.91324
参数设置:
{'num_thread': 4, 'num_leaves': 81, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08634000000000001
accuracy score:
0.91324
参数设置:
{'num_thread': 4, 'num_leaves': 81, 'metric': 'mape',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
```

```
0.08634000000000001
accuracy score:
0.91324
参数设置:
{'num_thread': 4, 'num_leaves': 81, 'metric': 'binary',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
mean of error_rate :
0.08634000000000001
accuracy score:
0.91324
```

进一步stacking

定义一个投票方法，对模型训练结果进行融合，得出accuracy score。

首先对train_k_fold方法进行修改，保留test_pred的[0,1]值，并将其返回出来。

```
def train_k_fold2(self, k_fold, train_data, test_data,
params=None, drop_test_y=True, use_best_eval=True):
        acc_result = list()
        train_pred = np.empty(train_data.shape[0])
        test_pred = np.empty(test_data.shape[0])
        if drop_test_y:
            dtest = test_data.drop(columns=self.label)
        else:
            dtest = test_data

        models = list()
        pred_all = list()
        for train_id, eval_id in k_fold.split(train_data):
            train_df = train_data.loc[train_id]
            eval_df = train_data.loc[eval_id]
            self.train(train_df, eval_df, params, use_best_eval)
            models.append(copy.deepcopy(self.clf))
```

```
            train_pred[eval_id] =
self.clf.predict(eval_df.drop(columns=self.label),
num_iteration=self.best_round)
            if self.metric == 'auc':
                y_pred =
self.clf.predict(eval_df.drop(columns=[self.label]),
num_iteration=self.best_round)
            else:
                y_pred =
(self.clf.predict(eval_df.drop(columns=[self.label]),

num_iteration=self.best_round) > 0.5).astype(int)
            acc_result.append(self.get_loss(eval_df[self.label],
y_pred))
            # test_pred += self.clf.predict(dtest,
num_iteration=self.best_round)
            pred_int = (self.clf.predict(dtest,
num_iteration=self.best_round) > 0.5).astype(int)
            pred_all.append(pred_int)
        # test_pred /= k_fold.n_splits
        return train_pred, pred_all, acc_result, models
```

其次，在定义一个vote的方法。它对上一步的输出中的模型训练结果的每一行进行投票，得出每一行最终的[0,1]值，形成vote_result的投票结果集。

上一步pred_all输出结果的head观测

	model-1	model-2	model-3	model-4	model-5
0	1	1	1	1	1
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	0	0	0	0	0

vote_func的代码

```
def vote_func(vote_list):
    vote_df =
pd.DataFrame(vote_list ,columns=['vote']).vote.value_counts()
    dict_vote = {'vote': vote_df.index, 'counts': vote_df.values}
    df_vote = pd.DataFrame(dict_vote).sort_values(by='counts',
ascending=False)
```

```
    vote_result = df_vote.vote[0]
    return vote_result
```

重新对train2进行模型训练，用test2进行评估。

因为上一轮num_leaves为12和81的结果较好，所以本轮训练num_leaves in [12,81]，其他参数保持不变。

```
from sklearn.model_selection import KFold
kfold = KFold(n_splits=5)
for num_leaves in [12, 81]:
  for metric in ['l1', 'rmse', 'quantile', 'mape', 'binary']:
    params = {'num_thread': 4, 'num_leaves': num_leaves,
'metric': metric, 'objective': 'binary',
              'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
    print('参数设置: ')
    print(params)
    train_pred, test_pred, error_rate, models =
fitter.train_k_fold2(kfold, train2, test2, params = params)
    vote_arr = pd.DataFrame(test_pred, index=['model-1',
'model-2', 'model-3', 'model-4', 'model-5']).T
    print(vote_arr.head())
    vote_result = list()
    for i in range(len(vote_arr.values)):
        vote_result.append(vote_func(vote_arr.values[i]))
    print(len(vote_result))
    print('accuracy score: ')
    print(accuracy_score(test2['loan_status'],vote_result))
```

训练结果

```
参数设置:
{'num_thread': 4, 'num_leaves': 12, 'metric': 'l1', 'objective':
'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
   model-1  model-2  model-3  model-4  model-5
```

```
0         1         1         1         1         1
1         1         1         1         1         1
2         1         1         1         1         1
3         1         1         1         1         1
4         0         0         0         0         0
accuracy score:
0.91372
参数设置:
{'num_thread': 4, 'num_leaves': 12, 'metric': 'rmse',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
   model-1  model-2  model-3  model-4  model-5
0         1         1         1         1         1
1         1         1         1         1         1
2         1         1         1         1         1
3         1         1         1         1         1
4         0         0         0         0         0
accuracy score:
0.91372
参数设置:
{'num_thread': 4, 'num_leaves': 12, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
   model-1  model-2  model-3  model-4  model-5
0         1         1         1         1         1
1         1         1         1         1         1
2         1         1         1         1         1
3         1         1         1         1         1
4         0         0         0         0         0
accuracy score:
0.91372
参数设置:
```

```
{'num_thread': 4, 'num_leaves': 12, 'metric': 'mape',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
   model-1  model-2  model-3  model-4  model-5
0        1        1        1        1        1
1        1        1        1        1        1
2        1        1        1        1        1
3        1        1        1        1        1
4        0        0        0        0        0
accuracy score:
0.91372
参数设置:
{'num_thread': 4, 'num_leaves': 12, 'metric': 'binary',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
   model-1  model-2  model-3  model-4  model-5
0        1        1        1        1        1
1        1        1        1        1        1
2        1        1        1        1        1
3        1        1        1        1        1
4        0        0        0        0        0
accuracy score:
0.91372
参数设置:
{'num_thread': 4, 'num_leaves': 81, 'metric': 'l1', 'objective':
'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
```

```
   model-1  model-2  model-3  model-4  model-5
0        1        1        1        1        1
1        1        1        1        1        1
2        1        1        1        1        1
3        1        1        1        1        1
4        1        1        1        0        1
accuracy score:
0.91322
参数设置:
{'num_thread': 4, 'num_leaves': 81, 'metric': 'rmse',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
   model-1  model-2  model-3  model-4  model-5
0        1        1        1        1        1
1        1        1        1        1        1
2        1        1        1        1        1
3        1        1        1        1        1
4        1        1        1        0        1
accuracy score:
0.91322
参数设置:
{'num_thread': 4, 'num_leaves': 81, 'metric': 'quantile',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
   model-1  model-2  model-3  model-4  model-5
0        1        1        1        1        1
1        1        1        1        1        1
2        1        1        1        1        1
3        1        1        1        1        1
4        1        1        1        0        1
accuracy score:
0.91322
```

参数设置：

```
{'num_thread': 4, 'num_leaves': 81, 'metric': 'mape',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
```

	model-1	model-2	model-3	model-4	model-5
0	1	1	1	1	1
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	0	1

```
accuracy score:
0.91322
```

参数设置：

```
{'num_thread': 4, 'num_leaves': 81, 'metric': 'binary',
'objective': 'binary', 'num_round': 1500, 'learning_rate': 0.001,
'feature_fraction': 0.5, 'bagging_fraction': 0.8}
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
The minimum is attained in round 1501
```

	model-1	model-2	model-3	model-4	model-5
0	1	1	1	1	1
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	0	1

```
accuracy score:
0.91322
```

通过对结果观察，当num_leaves=12时，accuracy score=0.91372为最好结果。

最终测试集预测结果为0.91372