**Notebook**：https://github.com/LoosonWu/geek\_homework/blob/main/stacking\_final.ipynb

**Base line**: 0.90122

**加上衍生变量**continuous\_income\_expense\_ratio

考虑的点是：以收入和花费的比例来indicate可否loan

公式是：train\_x['continuous\_income\_expense\_ratio'] = (train\_x['continuous\_annual\_inc']/12)/(train\_x['continuous\_installment']+train\_x['continuous\_funded\_amnt']\*train\_x['continuous\_int\_rate']/100/12)

**结果正确率是**：0.9014

**Stacking** 采用的是2层stacking，第一层用5个模型，第二层简易地用了一个模型。

结果正确率是：0.955115

**代码如下**：

clfs = [RandomForestClassifier(n\_estimators=5, n\_jobs=-1, criterion='gini'),

RandomForestClassifier(n\_estimators=5, n\_jobs=-1, criterion='entropy'),

ExtraTreesClassifier(n\_estimators=5, n\_jobs=-1, criterion='gini'),

ExtraTreesClassifier(n\_estimators=5, n\_jobs=-1, criterion='entropy'),

GradientBoostingClassifier(learning\_rate=0.05, subsample=0.5, max\_depth=6, n\_estimators=5)]

**def** stacking(train\_x, train\_y, x\_predict, y\_predict, n\_folds):

*'''采用2级stacking'''*

dataset\_blend\_train = np.zeros((train\_x.shape[0], len(clfs)))

dataset\_blend\_test = np.zeros((train\_y.shape[0], len(clfs)))

skf = StratifiedKFold(n\_splits=n\_folds)

*'''第一级stacking里有5个model'''*

**for** j, clf **in** enumerate(clfs):

dataset\_blend\_test\_j = np.zeros((train\_y.shape[0], n\_folds))

**for** i, (train, test) **in** enumerate(skf.split(train\_x,train\_y)):

print('i=',i)

X\_train, y\_train, X\_test, y\_test = train\_x[train], train\_y[train], train\_x[test], train\_y[test]

clf.fit(X\_train, y\_train)

y\_submission = clf.predict\_proba(X\_test)[:, 1]

dataset\_blend\_train[test, j] = y\_submission

*'''把每一个model的每一次split train起来的model用在测试数据集上测试后的结构放到dataset\_blend\_test\_j的一个column里'''*

dataset\_blend\_test\_j[:, i] = clf.predict\_proba(x\_predict)[:, 1]

*'''将一个model跑完后的预测结构求平均值作为这个model的结果'''*

dataset\_blend\_test[:, j] = dataset\_blend\_test\_j.mean(1)

print("val auc Score: **%f**" % roc\_auc\_score(y\_predict, dataset\_blend\_test[:, j]))

*'''第二级stacking里有1个model'''*

clf = GradientBoostingClassifier(learning\_rate=0.02, subsample=0.5, max\_depth=6, n\_estimators=30)

*'''将第一级stacking的结构用来拟合'''*

clf.fit(dataset\_blend\_train, train\_y)

*'''将测试数据集在第一级stacking里的预测值用来作为第二级stacking模型的测试级'''*

y\_submission = clf.predict\_proba(dataset\_blend\_test)[:, 1]

*# y\_submission = (y\_submission - y\_submission.min()) / (y\_submission.max() - y\_submission.min())*

print("val auc Score: **%f**" % (roc\_auc\_score(y\_predict, y\_submission)))