MUFG Data Science Basic Camp 2024 1st place solution

名古屋大学情報学部自然情報学科3年
2024年9月24日　壁谷　悠成

目次

01
自己紹介

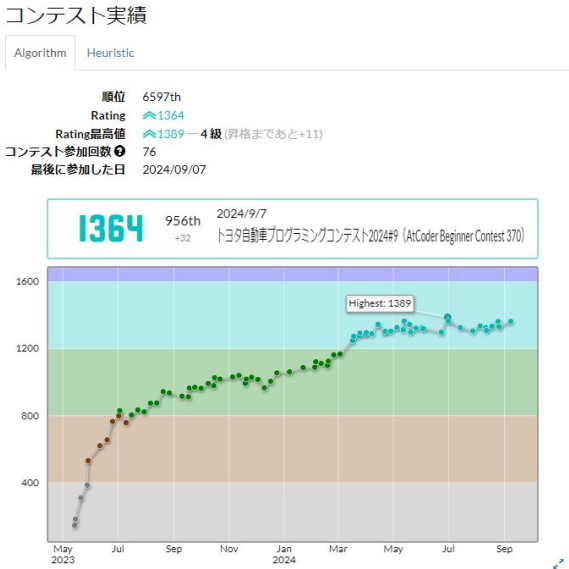
自己紹介

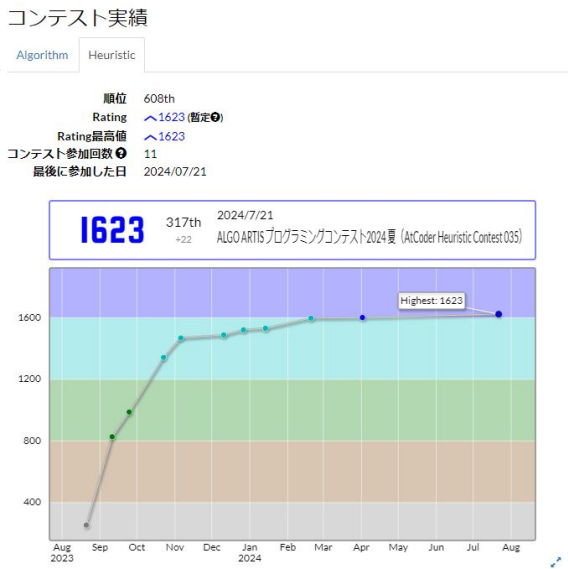
壁谷　悠成

名古屋大学情報学部自然情報学科 3年

趣味
・競技プログラミング
　Atcoder(水, 青)

X @melo_atc





02
解法の紹介

振り返って大事だったと思うこと

- **CV Score**
  - 暫定評価のScoreがCV Score + 0.005 ~ 0.008
  - 提出ファイルが1個しか選べない
  - 最終的にはCV Scoreを見て選択ファイルを決めた

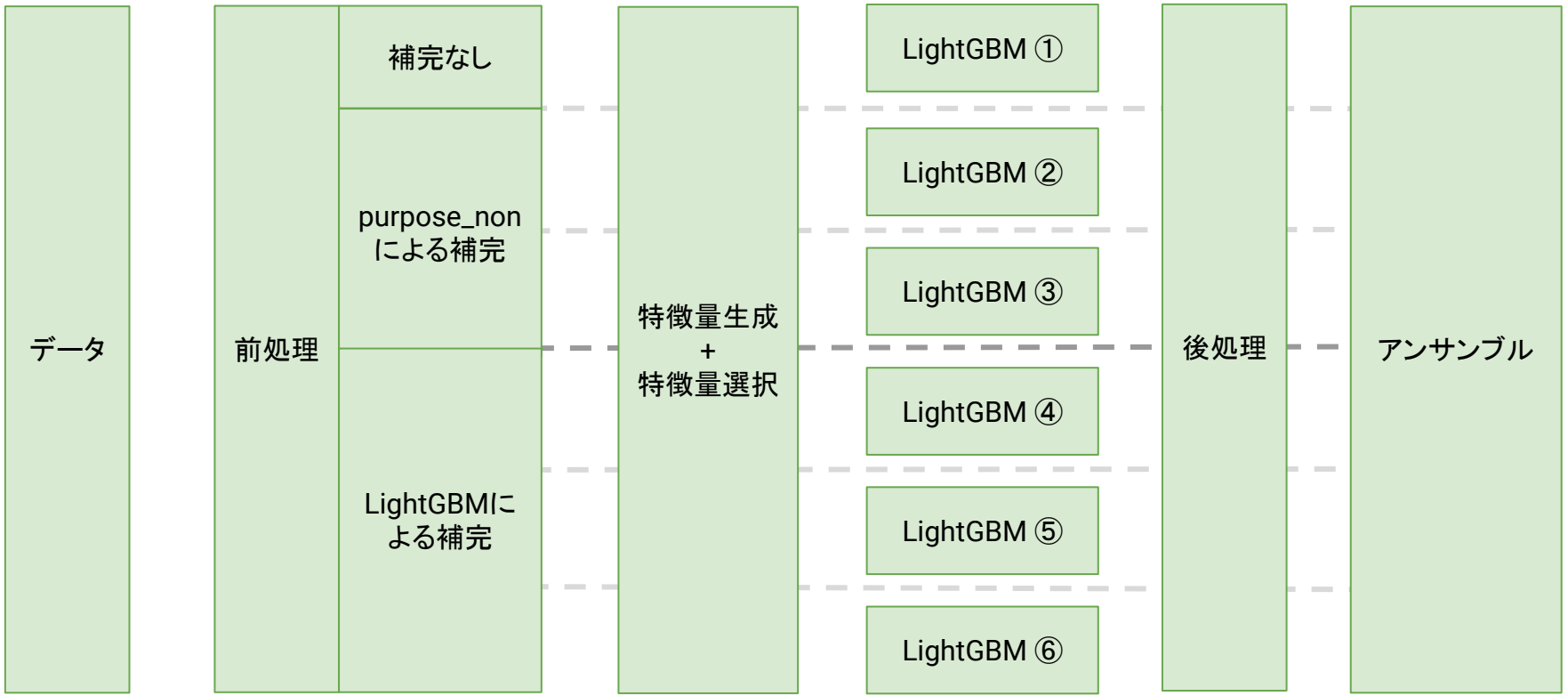
- **機械的な特徴量選択**
  - 自分で考えて生成した特徴量はすべて精度低下につながった

- **思いついたアイディアを実装し続けた**
  - 質よりも量を重視して取り組んだ結果うまくいった

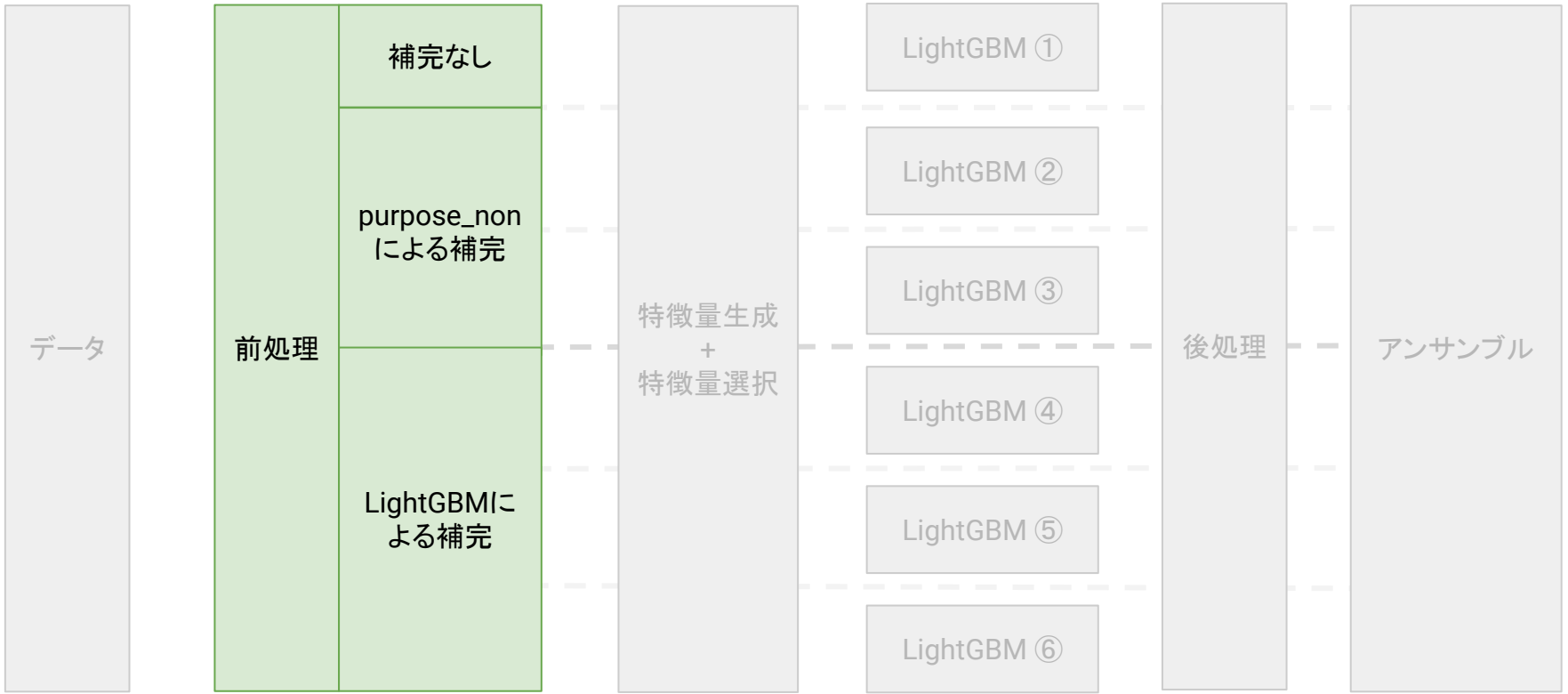
振り返って大事だったと思うこと

提出日	CV Score	暫定評価	最終評価	内容	最終評価順位
8 / 21	0.7944	0.8032	0.7948	自分で考えた特徴量の追加	4位相当
9 / 2	0.7951	0.8032	0.7943	機械的に特徴量を選択した後、人力で特徴量を選別	9位相当
9 / 4	0.7968	0.8028	0.7952	アンサンブルをしたうちの 1つのファイル	1位相当
9 / 5	0.7980	0.8035	0.7954	6つのファイルのアンサンブル	1位相当

全体像


データ
前処理
補完なし
purpose_non
による補完
LightGBMに
よる補完
特徴量生成
+
特徴量選択
LightGBM ①
LightGBM ②
LightGBM ③
LightGBM ④
LightGBM ⑤
LightGBM ⑥
後処理
アンサンブル

前処理


データ
前処理
補完なし
purpose_non
による補完
LightGBMに
よる補完
特徴量生成
+
特徴量選択
LightGBM ①
LightGBM ②
LightGBM ③
LightGBM ④
LightGBM ⑤
LightGBM ⑥
後処理
アンサンブル

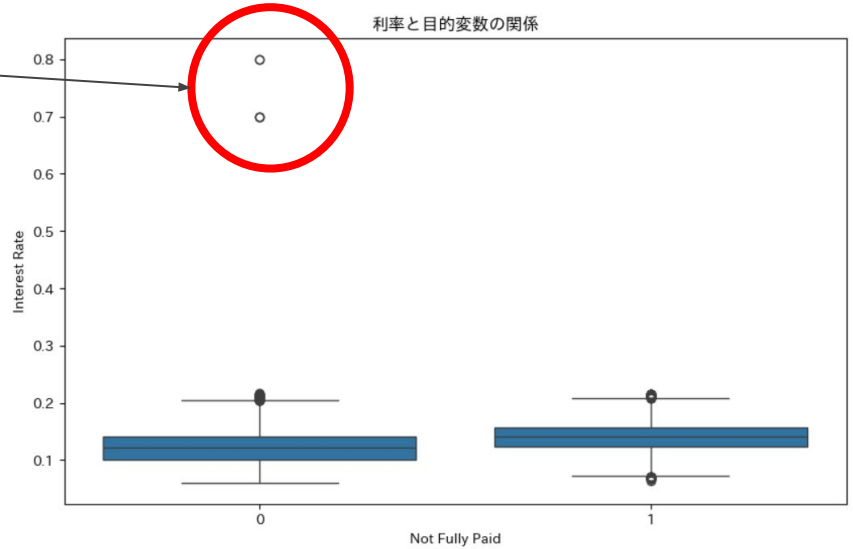
前処理

- **データの概要**
  - 欠損値や外れ値がほぼない、カテゴリ変数は1つで表記揺れなし
  - 前処理でできることはあまりない
- **欠損値補完について**
  - purpose以外の欠損値を持つ行は削除した
  - purposeについては補完無し、purpose_nonによる補完
    LightGBMの予測値による補完の3種類を試した

欠損値がある特徴量	欠損値の数(train.csv)	欠損値の数(test.csv)
purpose	773	504
installment	6	4
revol.bal	1	6
revol.util	12	17

前処理

- **外れ値除去について**
  - int.rateが0.5以上の行を削除した



- **カテゴリ変数について**
  - purposeをTarget Encodingにより数値に変換した

特徴量生成 + 特徴量選択


データ
前処理
補完なし
purpose_non
による補完
LightGBMに
よる補完
特徴量生成
+
特徴量選択
LightGBM ①
LightGBM ②
LightGBM ③
LightGBM ④
LightGBM ⑤
LightGBM ⑥
後処理
アンサンブル

特徴量生成

- **時間データの処理**
  - days.with.cr.lineを半年、1年、2年間隔で区切ったものを新しい特徴量として追加した

- **aggregated features**
  - purposeやdelinq.2yrsのそれぞれの値に対するint.rateやficoの max, min, max-min, std, mean, median, quantile(0.25), quantile(0.75)を新しい特徴量として追加した

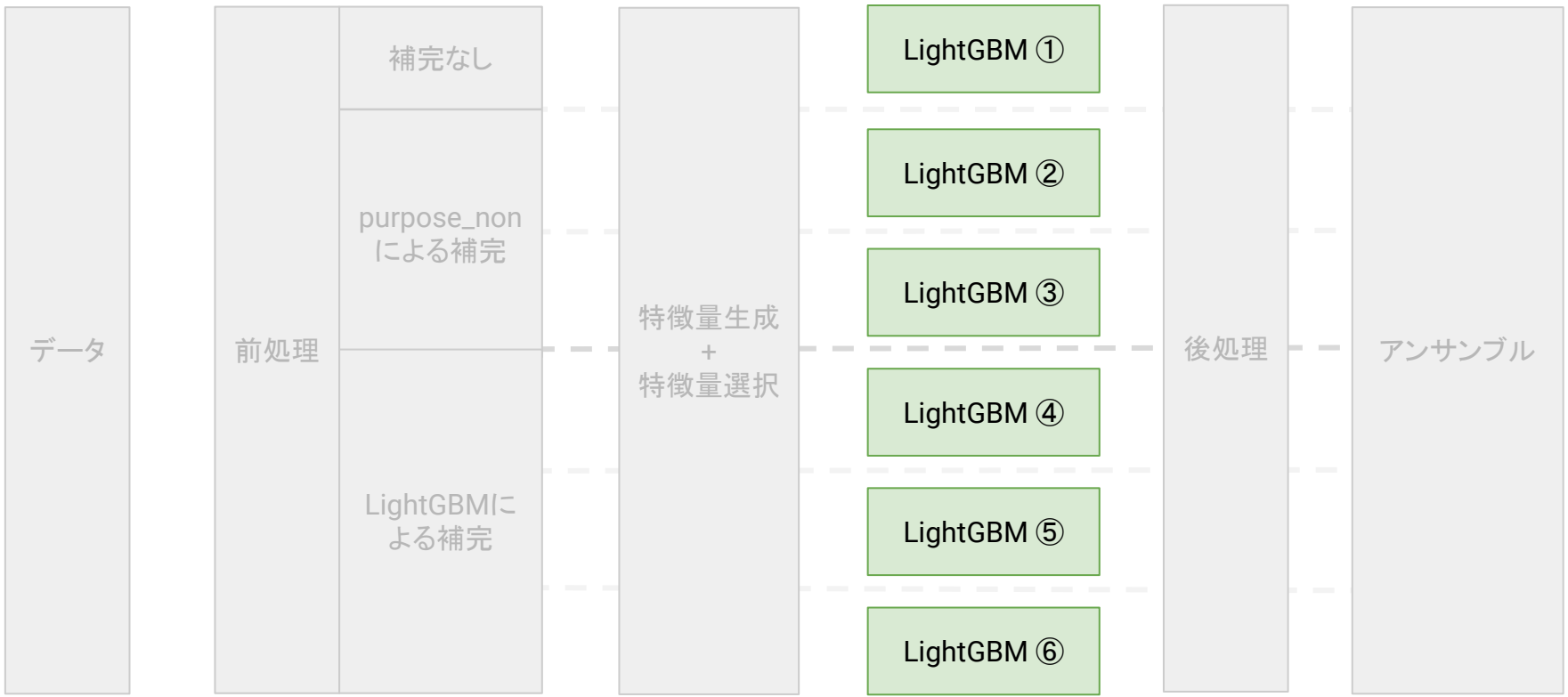
- **変数同士の掛け算と割り算**
  - 特徴量同士の掛け算と割り算をしたものを新しい特徴量として追加した

特徴量選択

- **特徴量選択①**
  - KS検定によりtrainとtestで分布が異なる特徴量を削除

- **特徴量選択②**
  - 元々の特徴量と生成した特徴量を分け、それぞれをシャッフルした後1つずつ変数を追加し、StratifiedKFold (n=5) でのLightGBMのScoreが上がった場合は採用し、下がった場合は削除する
  - StratifiedKFoldのrandom_stateについては1から1000の中で各CVでのScoreの分散が少ないseedを使用した

LightGBMによる予測



LightGBMによる予測

- **ハイパーパラメータ**
  - Optunaを用いてCV Scoreが最大となるようにチューニング
  - CV Scoreについては特徴量選択の際と同じくStratifiedKFold (n=5)により求めた
  - チューニングするハイパーパラメータについては、自分が重要そうだと思ったパラメータをすべて追加した

- **予測**
  - チューニングしたパラメータを用いて予測を行った

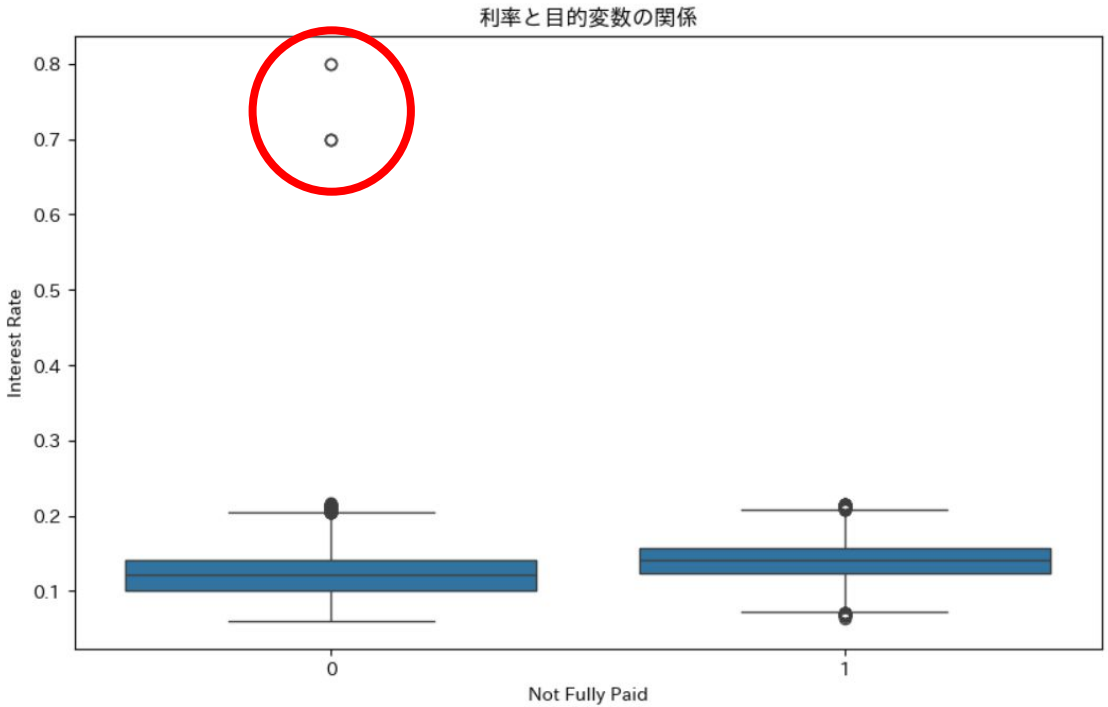
```
params = {
    'objective': 'binary',
    'metric': 'auc',
    'extra_tree': True,
    'boosting_type': 'gbdt',
    'n_estimators': 99999,
    'seed': SEED,
    'learning_rate': 0.020885306641593653,
    'num_leaves': 32,
    'colsample_bytree': 0.6031758618507362,
    'subsample': 0.9445306340392152,
    'max_depth': 3,
    'reg_alpha': 3.5882262964892297,
    'reg_lambda': 0.592022145325183,
    'min_child_weight': 0.0834484551537235,
    'feature_fraction': 0.4730667204518139,
    'bagging_fraction': 0.9423011633687252,
    'bagging_freq': 3,
    'verbosity': -1
}
```

後処理


データ
前処理
補完なし
purpose_nonによる補完
LightGBMによる補完
特徴量生成
+
特徴量選択
LightGBM ①
LightGBM ②
LightGBM ③
LightGBM ④
LightGBM ⑤
LightGBM ⑥
後処理
アンサンブル

後処理

- **int.rateによる判断**
  - int.rateが0.5以上の行の予測値を0にした



アンサンブル


データ
前処理
補完なし
purpose_nonによる補完
LightGBMによる補完
特徴量生成
+
特徴量選択
LightGBM ①
LightGBM ②
LightGBM ③
LightGBM ④
LightGBM ⑤
LightGBM ⑥
後処理
アンサンブル

アンサンブル（加重平均）

各ファイルのCV Scoreに基づいて適当に重みを決めた後、上から順に重みを0から0.5まで0.01刻みで、アンサンブル後のCV Scoreが最大化するように調整

ファイル名	CV Score	重み
LightGBM ①	0.7971	0.23
LightGBM ②	0.7972	0.26
LightGBM ③	0.7969	0.19
LightGBM ④	0.7968	0.18
LightGBM ⑤	0.7966	0.15
LightGBM ⑥	0.7963	0.13
アンサンブル結果	0.7980	

03
まとめ

自分の解法の改善点

- **特徴量選択の際のリーク**
  - 特徴量選択をCV全体でやっているためリークが発生している
  - 特徴量選択は各CV分割毎にやる必要がある
- **他のモデルの使用**
  - CatBoostなどの他のモデルをアンサンブルに加えることにより更なる精度向上が見込める
- **ハイパーパラメータ**
  - 知識がなかったので、他のコンペでの解法記事を読んで、チューニングをするパラメータを決めた
  - それぞれのパラメータの意味をあまり理解していない

試したけど上手くいかなかったこと

- **特徴量生成**
  - 遺伝的アルゴリズム
  - 近傍500個でのnot.fully.paidの平均
  - 欠損値であるかのフラグ
  - 自分で考えて追加した特徴量

- **特徴量選択**
  - Boruta
  - Null Importance
  - Ridge回帰による前進選択

楽しいコンペをありがとうございました！！