SIGNATE Cup 2024

～3rd Place Solution～

2024 / 9 / 21

Y H-poro（ほかぞの）

発表内容

1　自己紹介

2　解法紹介

（1）前処理

（2）試行と方針検討

（3）モデル構築

3．まとめ

2　解法紹介　（1）データ前処理

［データ前処理のフロー］

説明変数			1．データ表現の統一				2．数値化		3.欠損値の補完	4.標準化
			表記統一	誤字等修正	単位統一	分割表記統	カテゴリ変数化	文字列数値化		
1	Age		○					○	○	○
2	TypeofContact						○		○	
3	CityTier									
4	DurationOfPitch				○				○	○
5	Occupation						○			
6	Gender		○				○			
7	NumberOfPersonVisiting									
8	NumberOfFollowups			○					○	○
9	ProductPitched		○					○		
10	PreferredPropertyStar									
11	NumberOfTrips			○				○	○	○
12	Passport									
13	PitchSatisfactionScore									
14	Designation		○				○			
15	MonthlyIncome				○				○	○
16	customer_info	marital				○	○			
		car				○	○			
		chil				○		○	○	

・データ前処理は主にgooglecolabで実施
・カテゴリ変数はone-hotエンコーディング
・欠損値の補完は「IterativeImputer」で実施

2　解法紹介　（1）データ前処理

［最終的に分析に用いた説明変数］

	説明変数	カテゴリ変数	量的変数	
			連続変数	順序変数
1	Age		○	
2	TypeofContact	○		
3	CityTier			○
4	DurationOfPitch		○	
5	Gender	○		
6	NumberOfPersonVisiting		○	
7	NumberOfFollowups		○	
8	ProductPitched			○
9	PreferredPropertyStar			○
10	NumberOfTrips		○	
11	Passport	○		
12	PitchSatisfactionScore			○
13	MonthlyIncome		○	
14	car	○		
15	chil		○	
16	Occupation_Large Business	○		
17	Occupation_Salaried	○		
18	Occupation_Small Business	○		
19	Designation_AVP	○		
20	Designation_Executive	○		
21	Designation_Manager	○		
22	Designation_Senior Manager	○		
23	Designation_VP	○		
24	marital_divorced	○		
25	marital_married	○		
26	marital_single	○		
27	marital_unmarried	○		

目的変数	ProdTaken（0:不成約　1:成約） ※不均衡データ（0:2992,1:497）
説明変数	27変数 16変数：カテゴリ変数（one-hot） 7変数：連続変数 4変数：順序変数
評価指標	AUC

［その他前処理での試行］

・age、monthly income等の階級化

・順序変数のone-hot化、ダミー変数化

・Occupation、Designationの順序変数化

→最終的にはもともとのデータ記載方法をそのまま活かす形を採用（ProductPitchedのみ順序変数化）

2　解法紹介　（2）試行と方針検討

[アプローチ]

○不均衡データ対策としてオーバーサンプリングしたデータセットを数パターン作成

○説明変数を掛け合わせた特徴量（交互作用項）を新たに数パターン作成

手法の絞り込みも含めとりあえずpycaretで分析実施

［参考］負例：正例＝1:1、交互作用項はなしのpycaretの結果

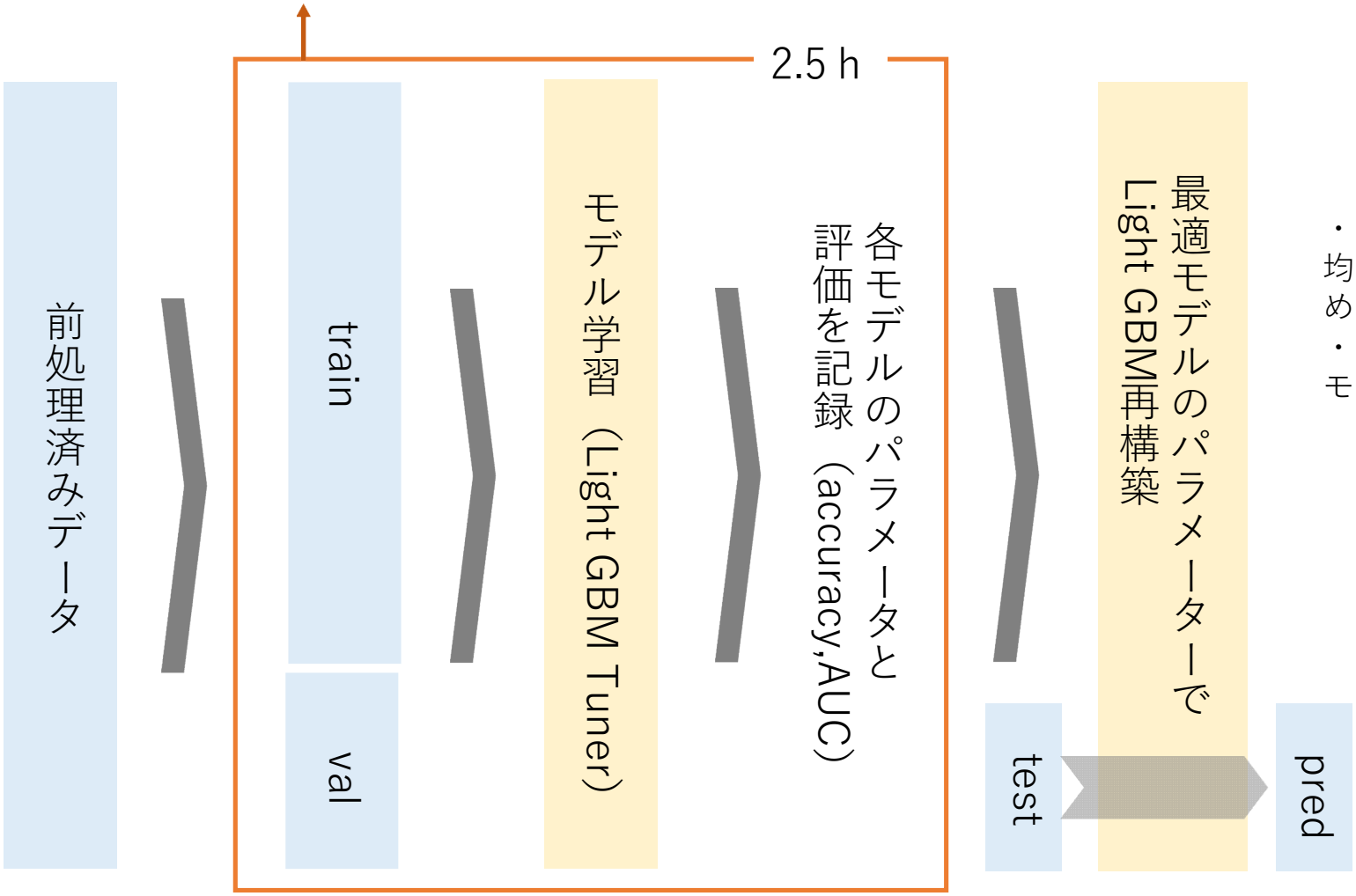
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
et	Extra Trees Classifier	0.9372	0.9828	0.9312	0.9427	0.9368	0.8744	0.8747
catboost	CatBoost Classifier	0.9341	0.9776	0.9145	0.9524	0.9328	0.8682	0.8694
rf	Random Forest Classifier	0.9320	0.9803	0.9355	0.9292	0.9322	0.8639	0.8642
lightgbm	Light Gradient Boosting Machine	0.9317	0.9743	0.9141	0.9481	0.9305	0.8634	0.8644
lr	Logistic Regression	0.9312	0.9733	0.8993	0.9610	0.9289	0.8625	0.8645
svm	SVM - Linear Kernel	0.9277	0.0000	0.8854	0.9678	0.9244	0.8553	0.8589
xgboost	Extreme Gradient Boosting	0.9277	0.9729	0.9107	0.9431	0.9264	0.8553	0.8562
ridge	Ridge Classifier	0.9272	0.0000	0.8763	0.9757	0.9231	0.8543	0.8591
lda	Linear Discriminant Analysis	0.9267	0.9729	0.8754	0.9757	0.9226	0.8534	0.8582
gbc	Gradient Boosting Classifier	0.9112	0.9652	0.9059	0.9159	0.9107	0.8224	0.8227
ada	Ada Boost Classifier	0.9002	0.9601	0.9012	0.8998	0.9003	0.8004	0.8006
dt	Decision Tree Classifier	0.8637	0.8637	0.8949	0.8428	0.8680	0.7273	0.7289
qda	Quadratic Discriminant Analysis	0.8574	0.8776	0.8673	0.8734	0.8636	0.7148	0.7265
knn	K Neighbors Classifier	0.8360	0.9541	0.9924	0.7564	0.8583	0.6719	0.7076
nb	Naive Bayes	0.7355	0.8708	0.9236	0.6722	0.7776	0.4709	0.5093
dummy	Dummy Classifier	0.4990	0.5000	0.4000	0.1995	0.2662	0.0000	0.0000

[方針検討]

→trainへの精度は上がるが、testを用いた提出データの精度向上は見られず、、、
→決定木系のアルゴリズムが上位（ロジスティック回帰も悪くはなさそう）
→既出のスコア程度の精度は見込めそう、まずは決定木ベースのモデルでパラメータチューニングするか
→前ページまでのデータでOptuna の LightGBM Tunerを使ったところ精度向上

2　解法紹介　（3）モデル構築

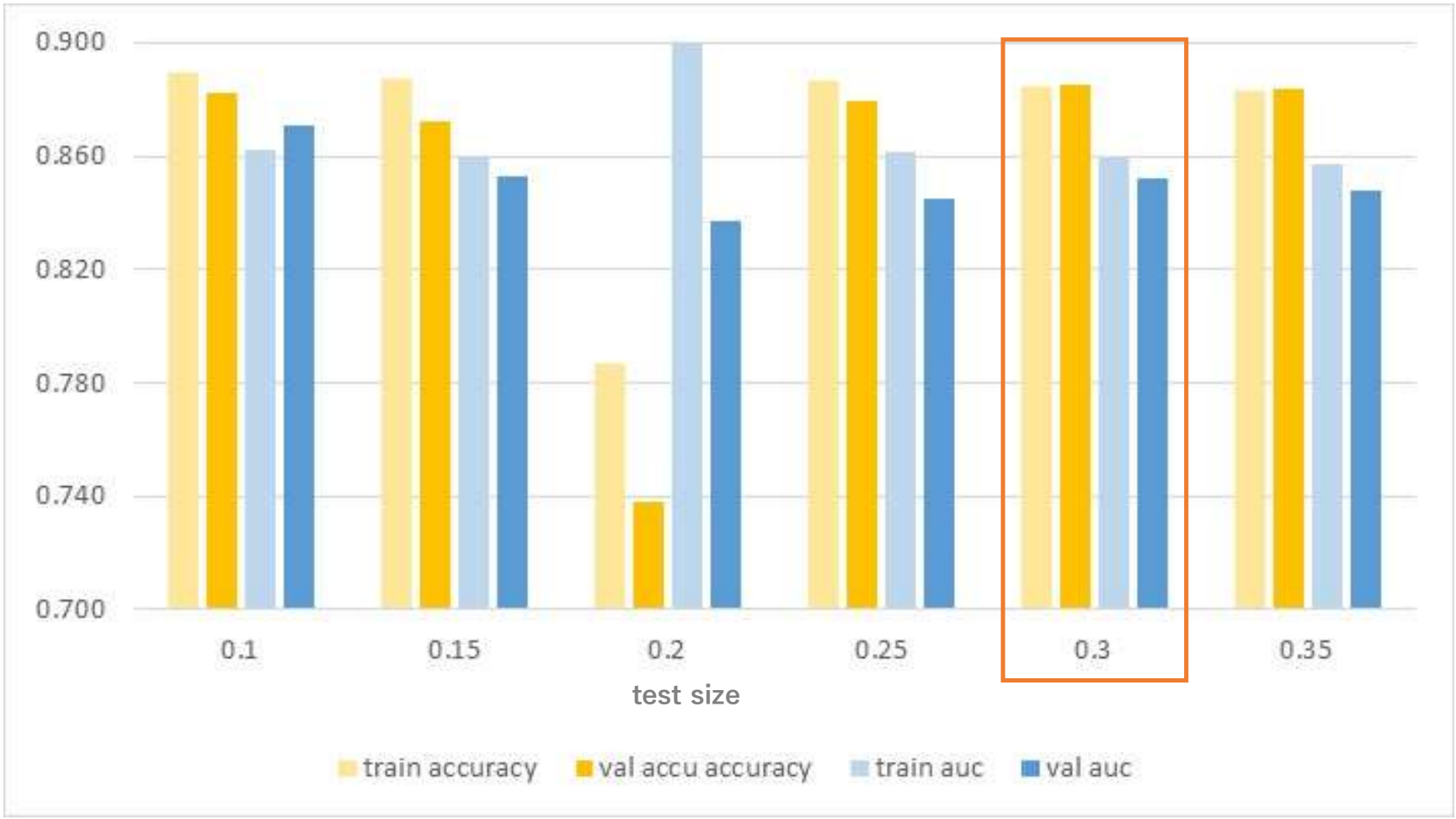
【Grid search】
・train_test_splitのtest_sizeを[0.1，0.15，0.2，0.25，0.3，0.35]
・lightgbmのmetric[auc,binary_error,binary_logloss] 及びboosting_type[gbdt,dart]
・その他のパラメータ最適化はoputunaに一任。



反省点
・CVで評価値の平均、ばらつき等も含めた評価
・シード値等によるモデルの再現性

2　解法紹介　（3）モデル構築

[モデル別の評価値グラフ（抜粋）]
※lightgbmのmetric[auc],boosting_type[dart]



3　まとめ

[反省点、感想等]

〇データ前処理や特徴量生成でもう少し効果的な工夫ができなかったか

〇他のモデルによるトライアルもできるとよかった

〇コンペに加え、コードの検収等を通じて、全体をまとめて俯瞰する視点やそこから見える改善点等を発見できた

主催の㈱signate様をはじめ協賛、後援いただいた
関係各社に感謝申し上げます