TabNet:
Attentive Interpretable Tabular Learning

読み会@2021/01/05

楊明哲

論文情報

- 著者
  - Sercan O. Arik, Tomas Pfister
  - Google Cloud AI
- 出典: ArxivのPreprint
  - ICLR 2020でリジェクトされた論文

概要

どんな論文？

- テーブルデータ向けのDNNモデル
- 決定木とNNモデルのいいとこ取りを目指した手法
  - 解釈性 + 精度 の向上が達成できた.

序論

研究背景

- DNNのモデルが特に画像,言語,音声の分野でSOTAである.
- Kaggleなどの分析コンペでは初めに決定木ベースの手法が主流
  - 解釈性が高いから

序論

研究背景

- なんでテーブルデータに対して，深層学習を取り入れたいのか？
  - 大規模なデータセットにたいしては，深層学習によって向上が期待できるから
  - Deep Learning Scaling is Predictable, Empirically.(Hestness et al., 2017)

序論

研究背景

- テーブルデータに対してNNモデルを使う3つのメリット

1. 複数のデータを効率よくエンコーディングできる
2. 特徴量エンジニアリングの手間を減らせる
3. End-to-endで扱うことができる.

序論

提案手法の貢献

- データの前処理を行わずにend-to-endでの学習を行える.
- 逐次注意を用いることで解釈性の高いモデルになっている.
  - Local interpretability: 入力特徴の重要度
  - Global interpretability: 各特徴量がモデルに対してどのくらい影響したか

関連研究

- DNN+DT
  - 逐次注意を用いて，特徴選択を行い特徴を入れ込んでいる.
- Tree-based learning
  - 特徴選択にDNNを用いている.
- Feature Selection
  - コンパクトな表現ができた.

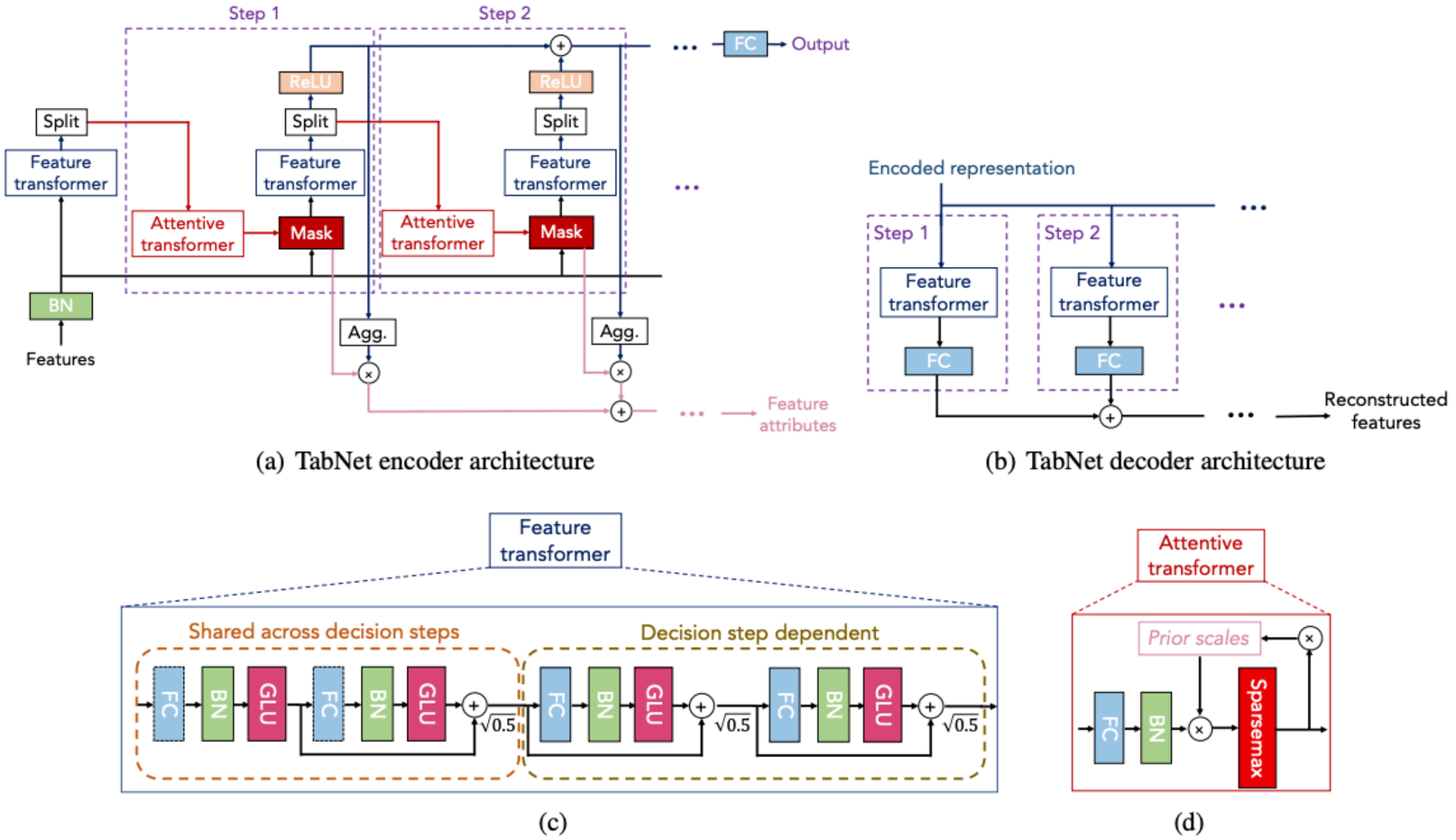
提案手法

重要なパーツ

- Attentive transformer
  - 特徴量に対して使うMaskの学習を行う.
- Feature transformer
  - 特徴量の変換，次ステップに使うものを決める.

提案手法

全体の構造



(a) TabNet encoder architecture

(b) TabNet decoder architecture

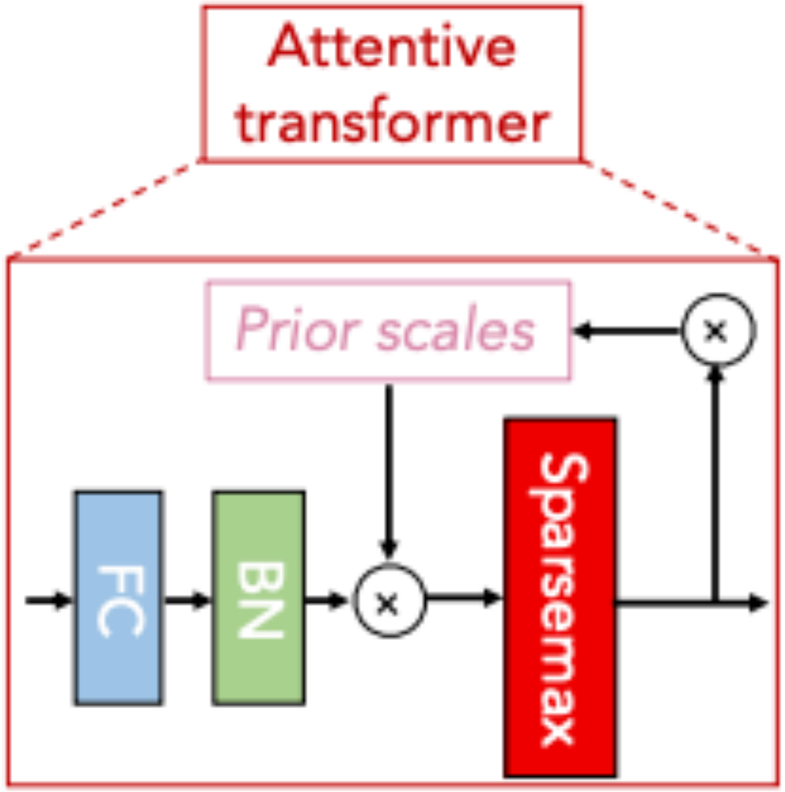
(c)

(d)

- これ以降出てくる i はステップ1,2,...に対応している

提案手法

Attentive Transformer: マスクの学習を行う.

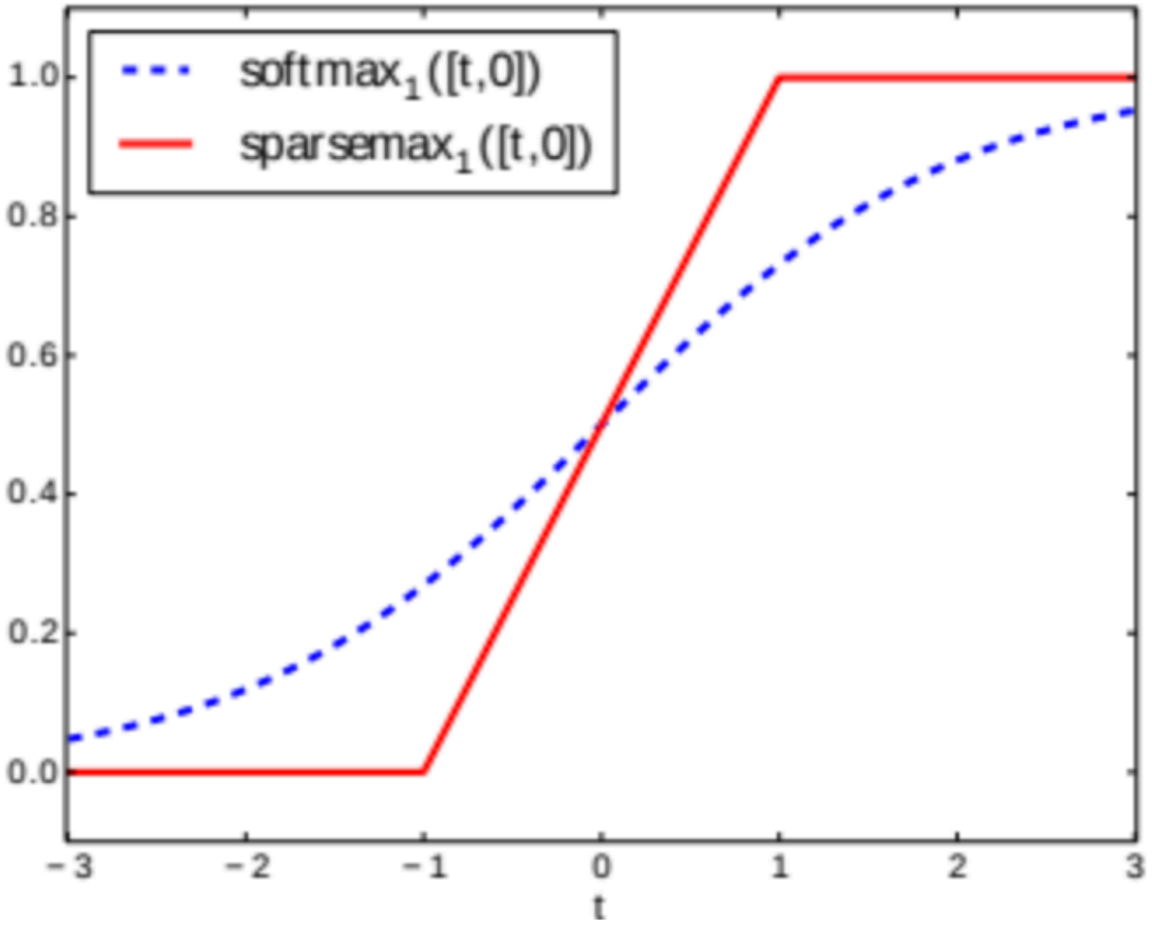


(d)

- $M[i] = \text{sparsemax}(P[i] \cdot h_i(a[i-1]))$
- $P[i]$: 過去のMで使われているか？によって変わる重み(実装では利用制限みたいなもの)
- Sparsemax: softmaxに似た活性化関数

コラム

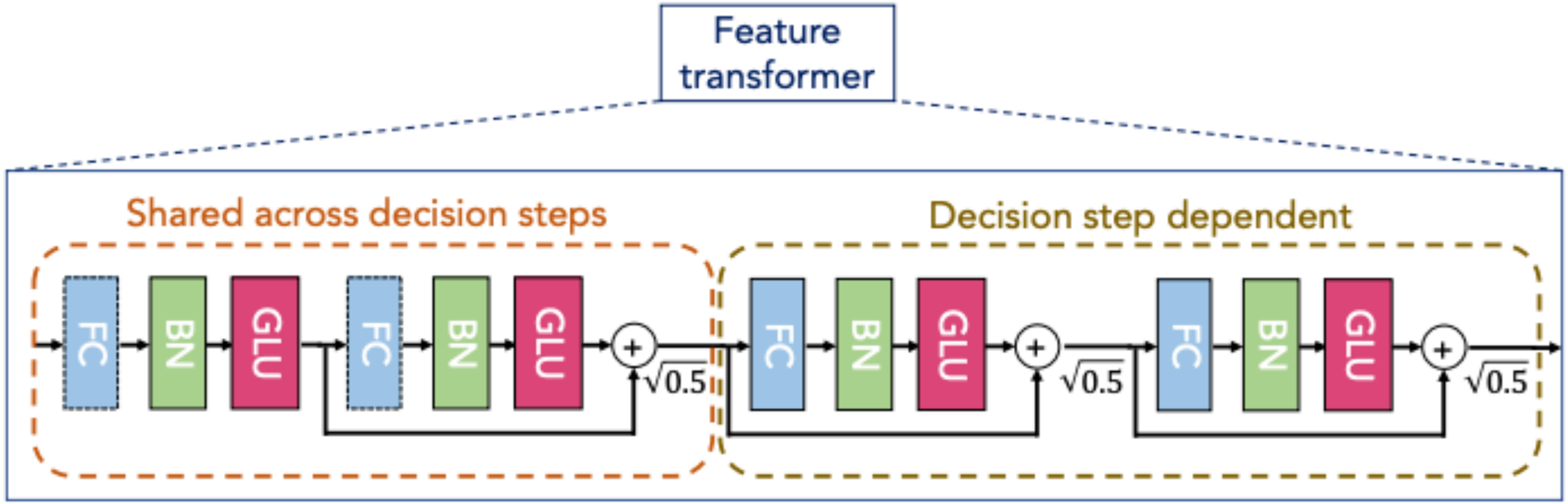
SparseMax (Andre et al., 2016)



- Softmaxよりも疎になりやすいから，重要な特徴量を取り出しやすい

提案手法

Feature Transformer: 入力を変換し，次に使うものを決める

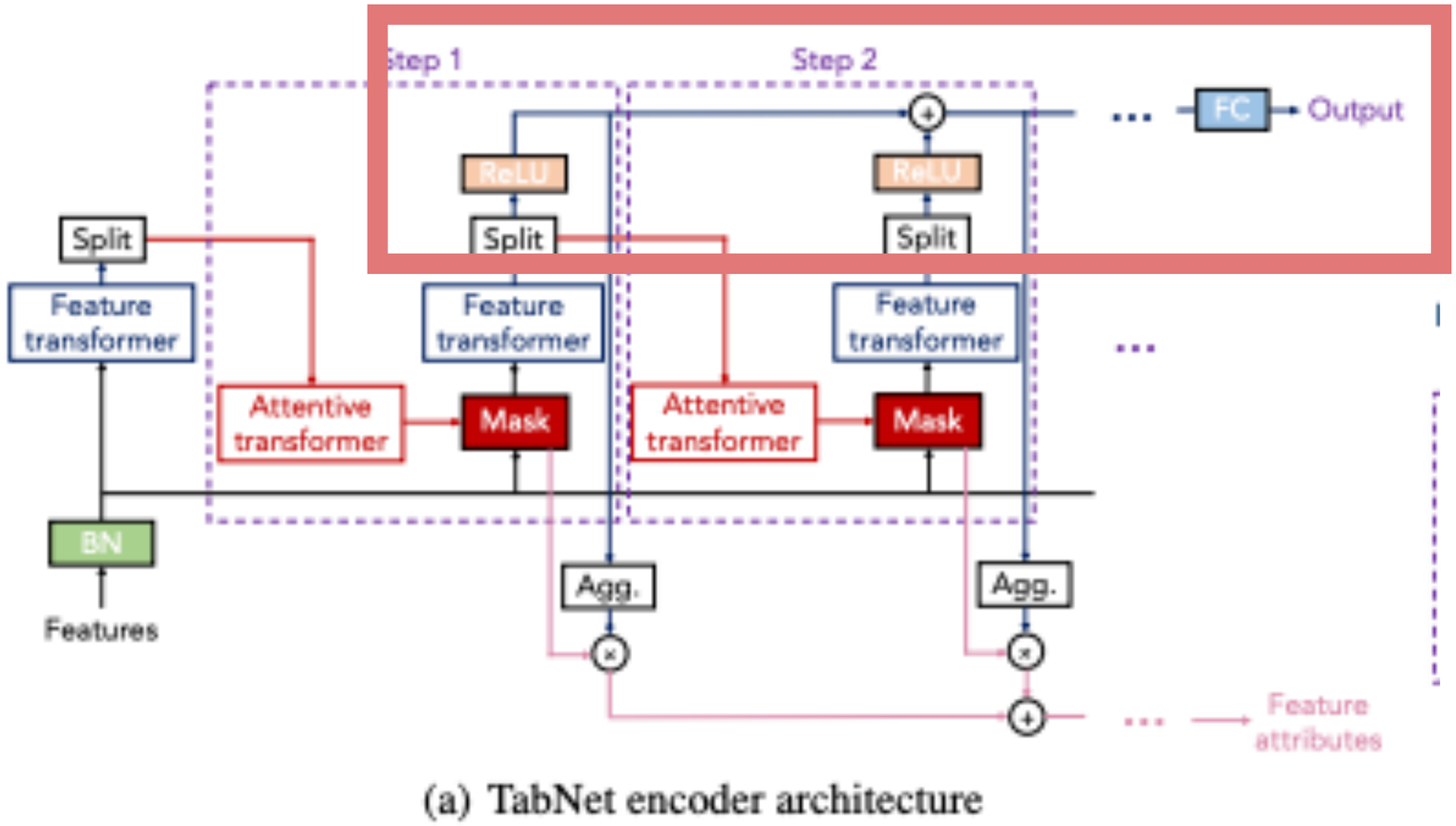


(c)

- $[d[i], a[i]] = f_i(M[i] \cdot f)$，aは次のステップに回される

提案手法

最終予測



(a) TabNet encoder architecture

- 各ステップ$d[i]$を集計して最終的な予測に用いる

提案手法

解釈性について

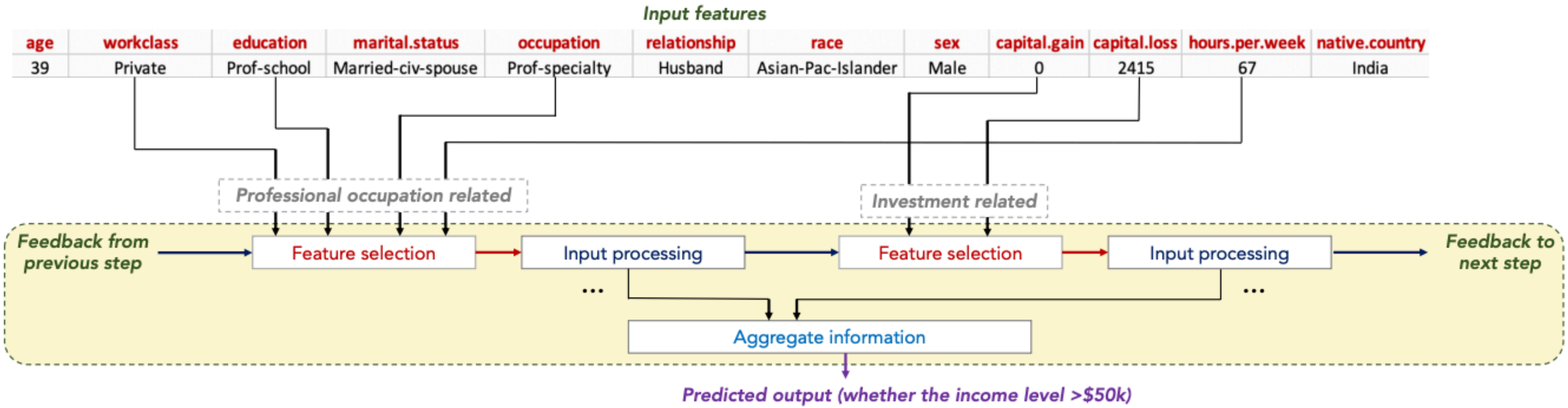
- 特徴量の重要度はマスクを使って計算する
  - 簡単に計算するため，マスクではなく特徴量を用いる

$$\eta_b[i] = \sum_{c}^{N_d} \mathrm{ReLU}(d_{b,c}[i]) \quad \rightarrow \text{どのサンプルが重要か？}$$

- $$M_{agg-b,j} = \frac{\sum_{i=1}^{N_steps} \eta_b[i] M_{b,j}[i]}{\sum_{j=1}^{D} \sum_{i=1}^{Nsteps} \eta_b[i] M_{b,j}[i]} \rightarrow \text{特徴量の重要度}$$

提案手法

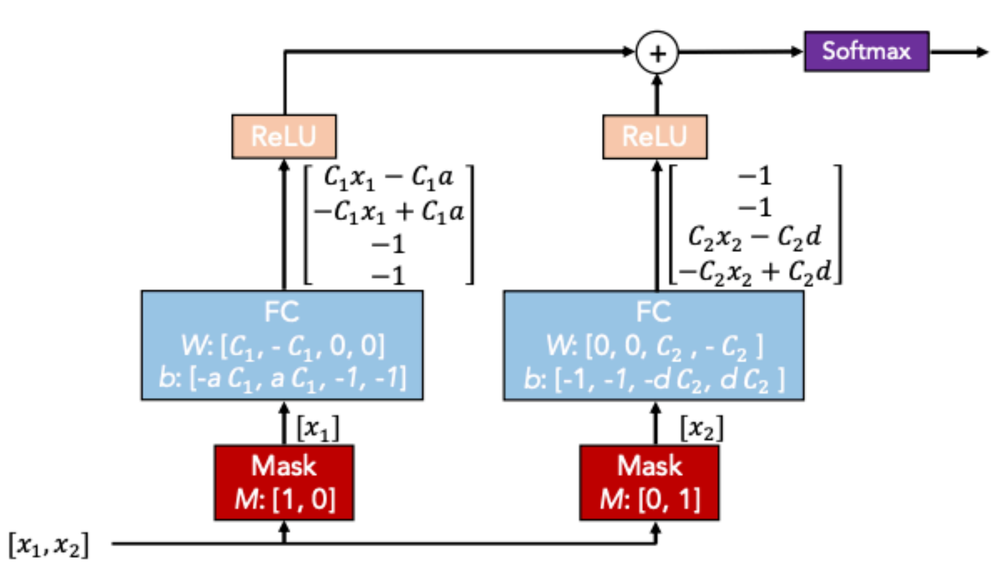
特徴量選択のイメージ



- Feature selectionが各ステップに対応

提案手法

どこが決定木ぽいの？



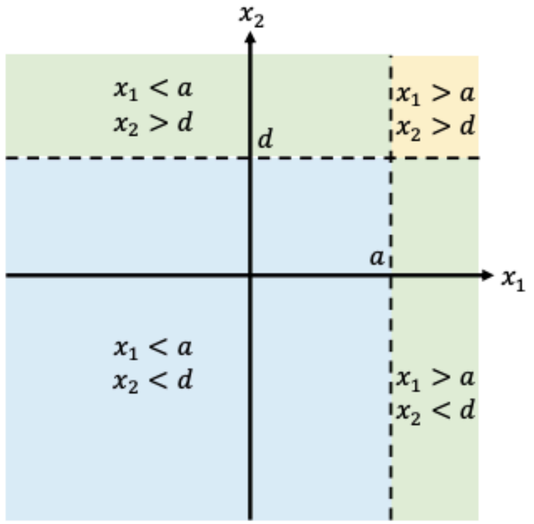


Figure 3: Illustration of DT-like classification using conventional DNN blocks (left) and the corresponding decision manifold (right). Relevant features are selected by using multiplicative sparse masks on inputs. The selected features are linearly transformed, and after a bias addition (to represent boundaries) ReLU performs region selection by zeroing the regions. Aggregation of multiple regions is based on addition. As C_1 and C_2 get larger, the decision boundary gets sharper.

- 各マスクによって作られる特徴量が分岐に対応している.

実験

実験設定

- 対抗手法:
  - 勾配ブースティング系: LightGBM, XGBoost, CatBoost
  - NNモデル
- なにで比べるか？
  - テストデータに対するaccuracy
  - モデルのサイズ

実験結果

精度に関して

Performance on real-world datasets

Table 2: Performance for Forest Cover Type dataset.

Model	*Test accuracy (%)*
XGBoost	89.34
LightGBM	89.28
CatBoost	85.14
AutoML Tables	94.95
TabNet	**96.99**

- 実データ(ForestCoverType)では対抗手法よりも精度が良かった.

実験結果

モデルサイズに関して

Model	*Test MSE*	*Model size*
Random forest	2.39	16.7K
Stochastic DT	2.11	28K
MLP	2.13	0.14M
Adaptive neural tree	1.23	0.60M
Gradient boosted tree	1.44	0.99M
TabNet-S	**1.25**	**6.3K**
TabNet-M	**0.28**	**0.59M**
TabNet-L	**0.14**	**1.75M**

Sarcos (Vijayakumar and Schaal 2000): The task is regressing inverse dynamics of an anthropomorphic robot arm.

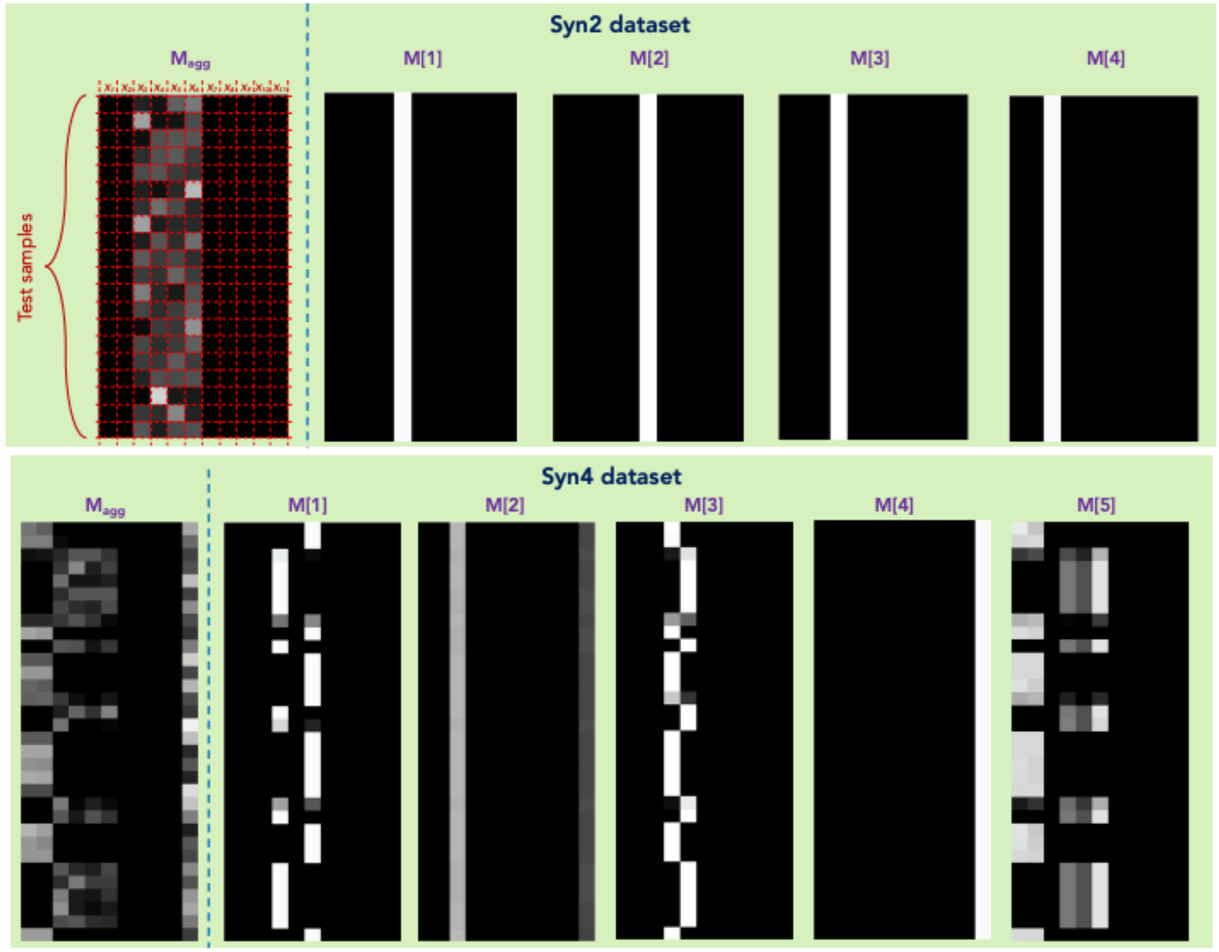
Model	*Test acc. (%)*	*Model size*
Sparse evolutionary MLP	**78.47**	**81K**
Gradient boosted tree-S	74.22	0.12M
Gradient boosted tree-M	75.97	0.69M
MLP	78.44	2.04M
Gradient boosted tree-L	76.98	6.96M
TabNet-S	78.25	81K
TabNet-M	**78.84**	**0.66M**

Higgs Boson (Dua and Graff 2017): The task is distinguishing between a Higgs bosons process vs. background. Due to

- モデルサイズが軽量でも精度がいい.

実験結果

解釈性について



- $\eta_b[i]$の結果を可視化
- 行がサンプル，列が特徴量
- 白いところが特徴量として重要と判断したところ

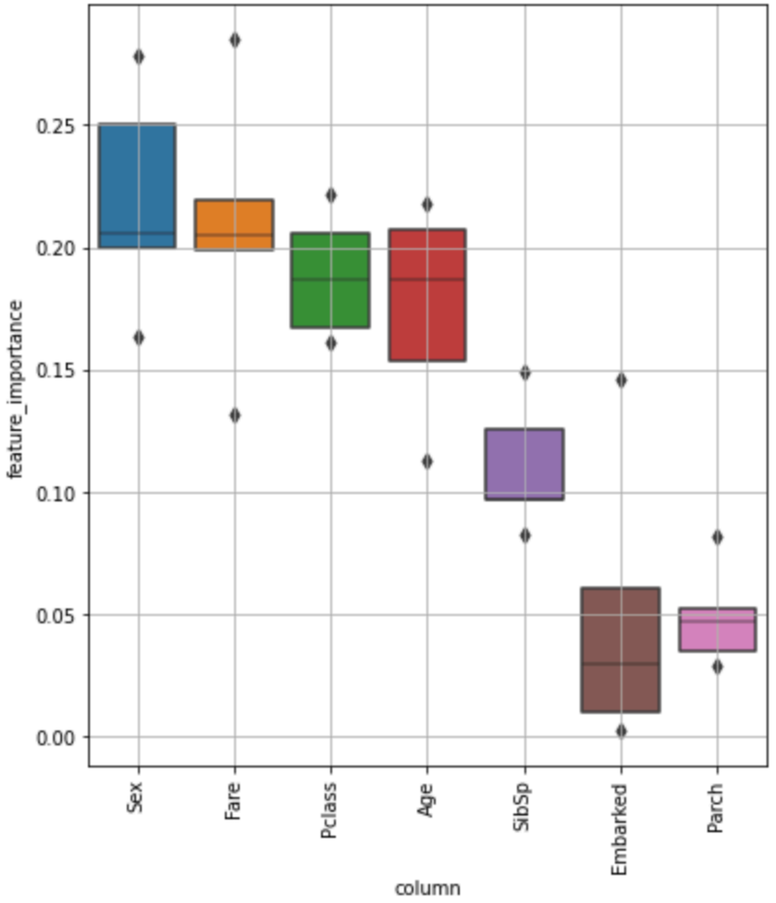
nce masks $\mathbf{M}[\mathbf{i}]$ (that indicate feature selection at i^{th} step) and the aggregate feature importance mask
l instance-wise feature selection, on Syn2 and Syn4 (Chen et al. 2018). Brighter colors show a higher

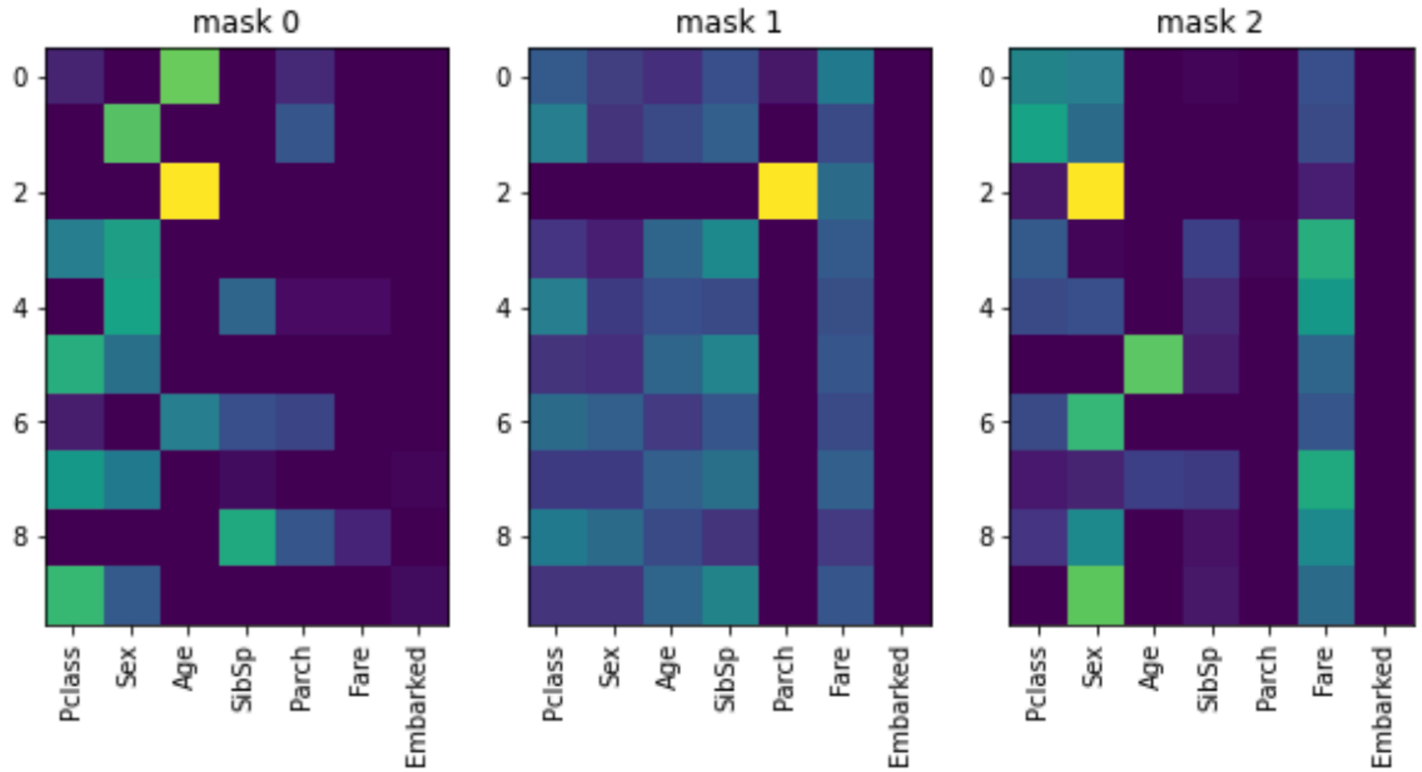
まとめ

- 逐次注意を行うことで，重要な特徴量選択を行なっている.
- マスクを用いることで解釈性の高いモデルになった.
- 様々な領域のテーブルデータでも性能を発揮できることを示した.

おまけ

TitanicデータセットでTabNetを遊んでみた.

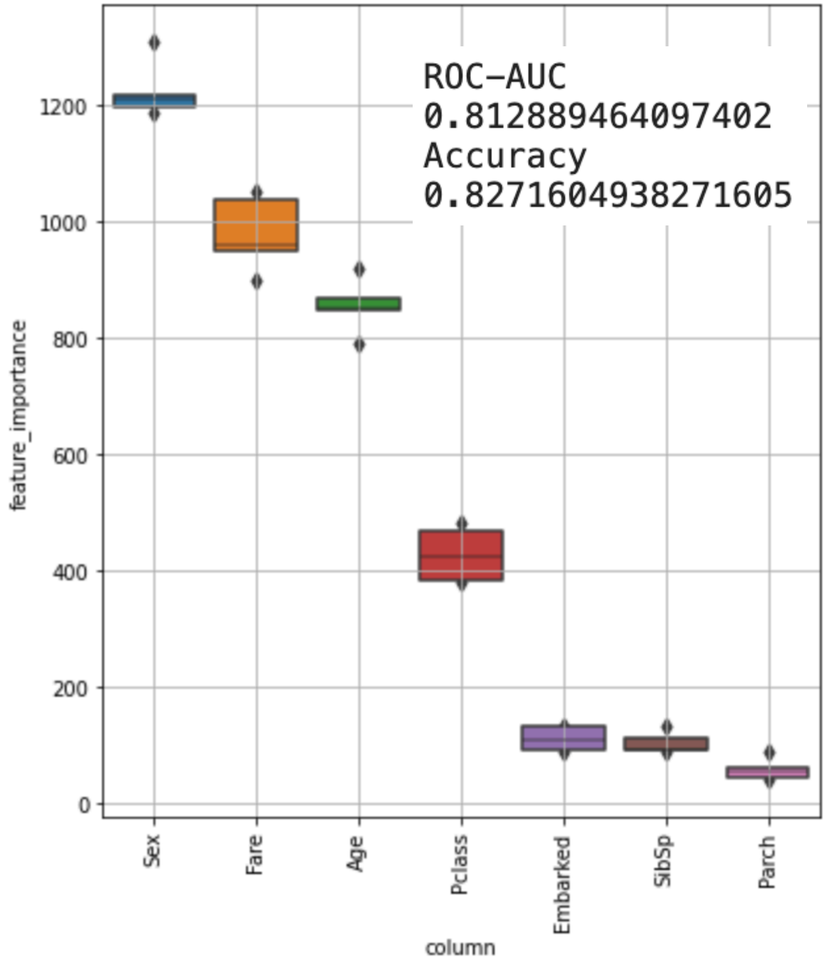




- Accuracy: 0.81, ROC-AUC: 0.78

おまけ

LightGBM vs NN model vs TabNet

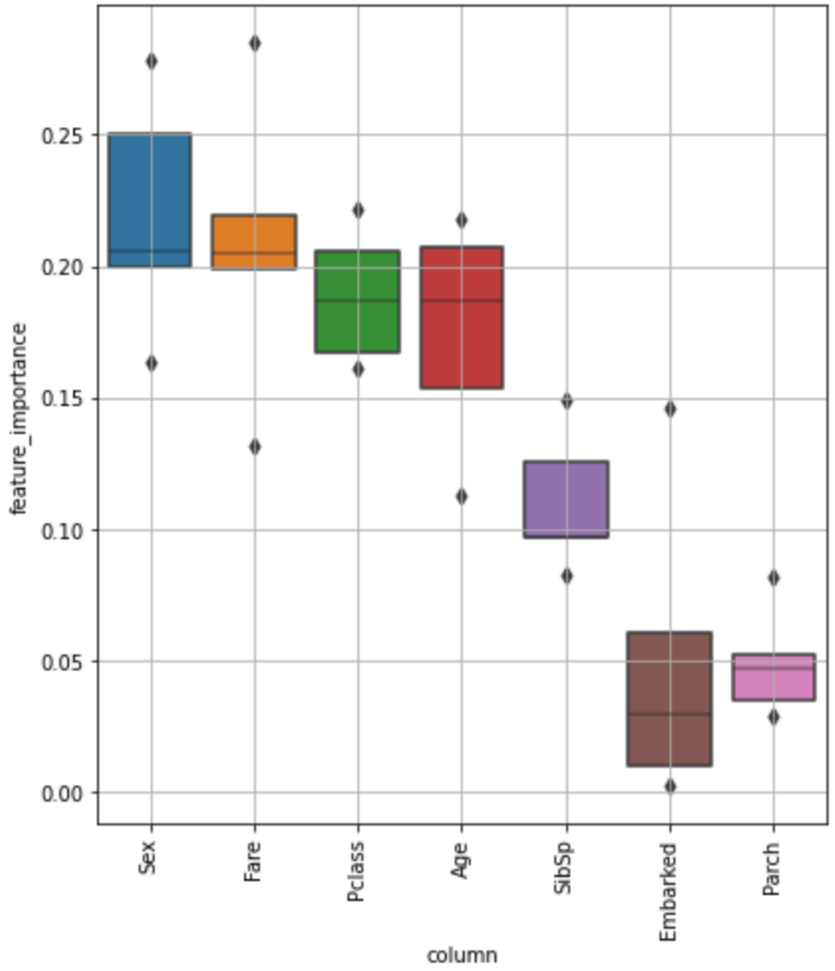


LightGBM

ROC-AUC
0.844818862578425 3
Accuracy
0.811447811447811 4

NN model

ハイパラは初期値のままで
チューニングを行なっていない



- **TabNet: Accuracy: 0.81, ROC-AUC: 0.78**

https://github.com/mei28/playground_python/blob/main/notebooks/titanic.ipynb