「技術スタッフ交流会プログラム」
データ構造化ワークショップ2024
Python中級者向け

ハイパーパラメータ 実践編

Smart Solutions株式会社

ハイパーパラメータとは (1)

- 機械学習プロセスの挙動を決定する、学習の前に人が決定しておくパラメータ
  - 大体のイメージとしてはこのような説明だが、もう少し正確に説明したい

機械学習の分野における「ハイパーパラメータ」の正確な定義は意外と難しい

- 日本語版Wikipediaには説明がない・・・英語版では・・・
  - "a parameter that can be set in order to define any configurable part of a model's learning process."

    訳：モデルの学習プロセスの設定可能な部分を定義するために設定できるパラメータ (つまり・・・どういうこと？？？)
- 大手企業の解説サイトより引用 (少しは分かりやすい・・・のか？)
  - AWS：データサイエンティストが機械学習モデルのトレーニングを管理するために使用する外部設定変数です。[1]
  - IBM：データサイエンティストが機械学習モデルのトレーニング・プロセスを管理するために事前に設定する構成変数のことです。[2]

[1] https://aws.amazon.com/jp/what-is/hyperparameter-tuning/
[2] https://www.ibm.com/jp-ja/think/topics/hyperparameter-tuning

ハイパーパラメータとは (2)

なるべく正確に、分かりやすく説明するならば

- **機械学習を行うプログラムの設定ファイルのようなイメージ**
  - 「機械学習を行うプログラム」とは、例えば scikit-learn が提供する RandomForestClassifier や SVC などの推定器(estimator)のこと
  - 実際には 設定"ファイル" ではなく、モデルを初期作成する時に引数として設定

```
model = RandomForestClassifier(
        n_estimators=200,
        max_depth=30
)
```
← ハイパーパラメータ

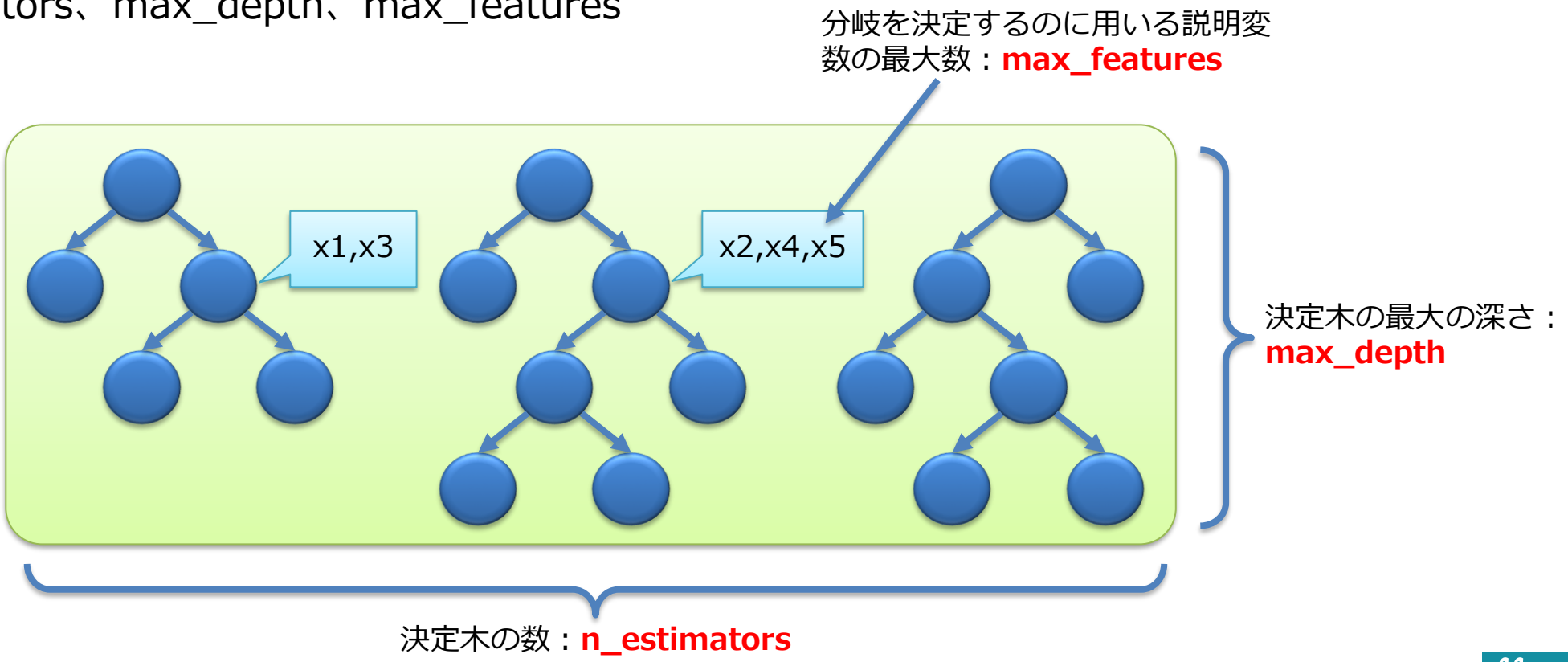
- 学習により値が決まるパラメータ(モデルパラメータ)とは全く異なるもの

パラメータ	決定方法	例
モデルパラメータ	学習の過程で自動的に変化	線形回帰：係数、切片 ランダムフォレスト：分割基準、葉ノードの値
ハイパーパラメータ	学習を始める前に人が決定 **→ 変更したら学習はゼロからやり直し！**	サポートベクタマシーン：正則化パラメータ、カーネル ランダムフォレスト：<次スライドにて紹介>

ハイパーパラメータの例

scikit-learnのランダムフォレスト推定器 RandomForestClassifier、RandomForestRegressor の主要なハイパーパラメータ

- n_estimators、max_depth、max_features



ハイパーパラメータの最適化

ハイパーパラメータの重要性

- モデルの学習プロセスや、最終的な性能に大きな影響を与える
- 適切に設定しないと、過学習や性能劣化が起こる可能性がある

ハイパーパラメータの決め方

- モデルの内部処理を見ても、論理的に値を決めることは難しい (まさにブラックボックス)
  - 多次元、かつ複雑な相互作用があり、非線形なパラメータも多い
- Optunaによるブラックボックス最適化が最適！！

ハイパーパラメータ 実践編 [Google Colaboratory]

https://colab.research.google.com/drive/1DJJTPb89kUao4d8TtAXgXVtFveUpLVAV?usp=sharing

パラメータ値の候補を決めるには (1)

- パラメータ値の候補数や、そもそものパラメータ数が多すぎると・・・
  - たとえベイズ最適化を使っても、満足のいく結果を得られるまで時間がかかりすぎる
  - 数時間どころか、数日かかるなんてことも

- 結局、無駄なパラメータ値候補やパラメータを減らして、探索範囲を必要最小限に減らすのが非常に有効
  - だけど・・・数を減らそうにも、どのくらいが適切か全然わからない (ブラックボックスなので！！)

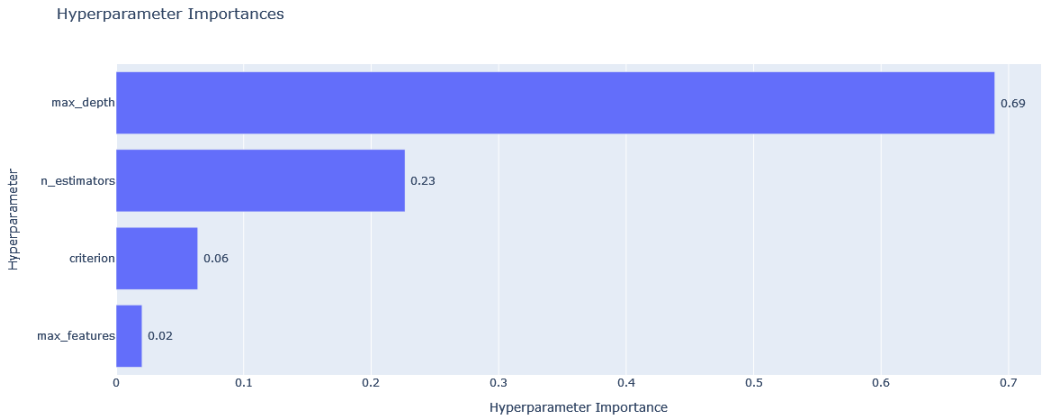
パラメータ値の候補・パラメータ数を減らすヒント 1

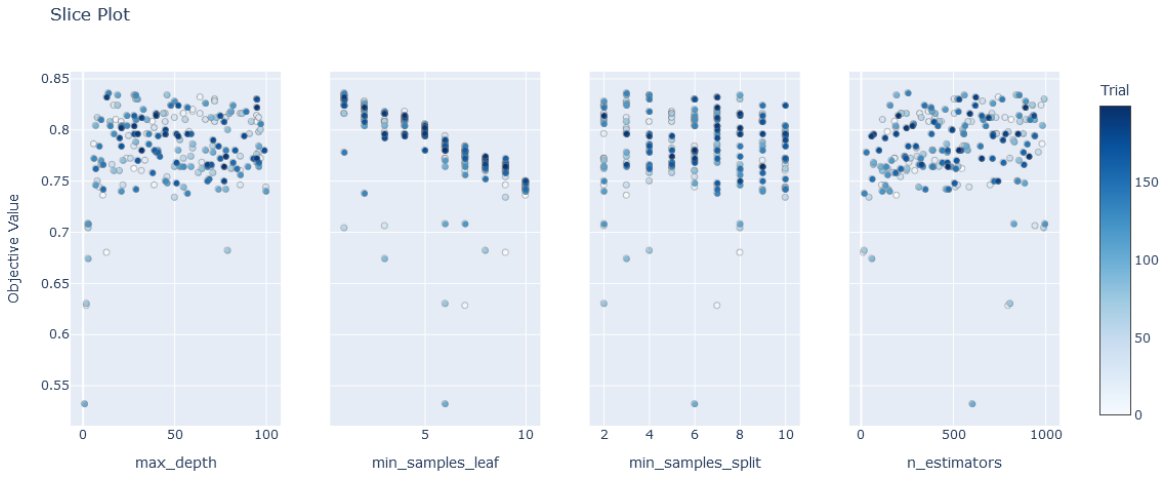
- scikit-learnのリファレンスなどでデフォルト値を調べて、そこから逸脱しない範囲をパラメータ値の候補に選ぶ
  - デフォルト値に選ばれるだけあり、多くのケースで優れていることが多い (専門家の知恵)

パラメータ値の候補を決めるには (2)

パラメータ値の候補・パラメータ数を減らすヒント 2

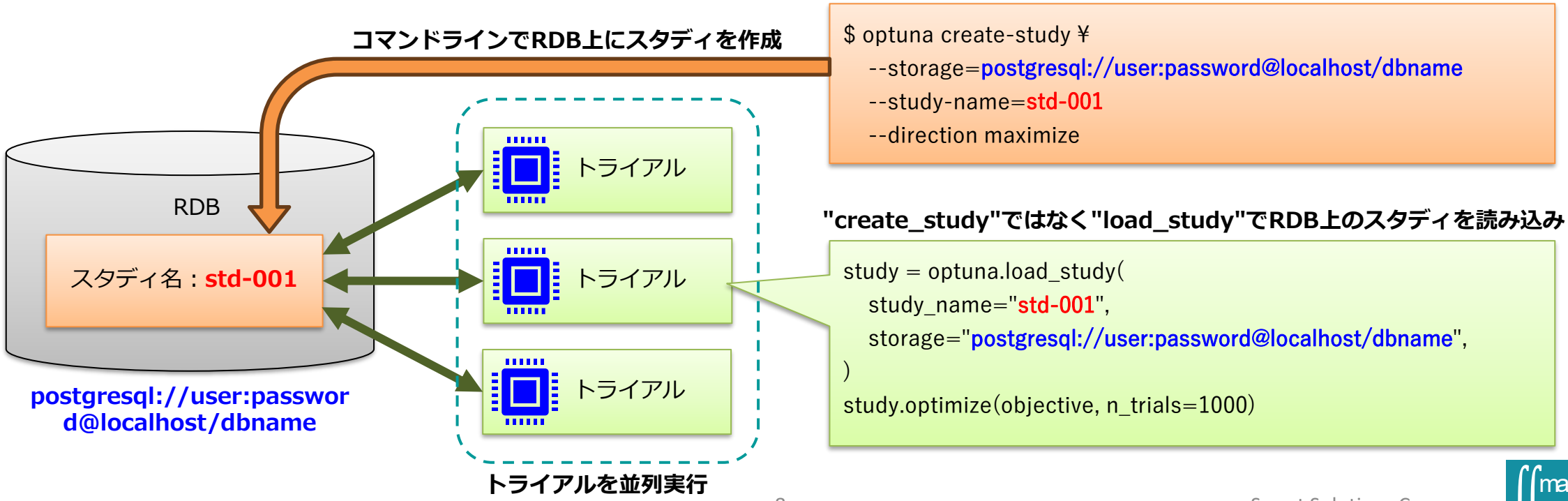
- ランダムサーチで十分なトライアルを試行し、パラメータ値と評価値の関係を分析する
  - 平均して評価値の高い、パラメータ値の範囲の見極め
  - パラメータ値と評価値の相関が極端に低いようであれば、最適化対象から外してデフォルト値を採用 (パラメータ数自体を減らしてしまう)
  - Google Colaboratoryの資料でご紹介した**plot_param_importances**メソッドや**plot_slice**メソッドを有効活用





[参考] 分散並列最適化

- 複数ノードを活用して、効率的にハイパーパラメータ探索を行う手法
  - Optunaの機能を活用して簡単に実行可能
  - トライアルを並列実行することで、探索スピードと精度が向上
  - トライアルの結果をRDBに集約し、各ノードがアクセスすることで並列化を実現



以上で、ハイパーパラメータ実践編の講義は終了です。

明日はいよいよ、グループワークでOptunaによる最適化にチャレンジします！！