問1

課題

covertype データセットの分類モデルを決定木を用いて作る。その際、グリッドサーチ / 交差検証を行い、ハイパーパラメータの一つである max_depth の最適な値を求める。

データセット

データセット (sklearn.datasets.fetch_covtype) を訓練データ (X_train, y_train) とテストデータ (X_test, y_test) に分割する。訓練データは最適なハイパーパラメータの探索（グリッドサーチの際に、さらに訓練データと検証データに分割される）とモデルの訓練に用い、テストデータは最終的に得られたモデルの予測に用いる。

結果 / 考察

決定木 (チューニング前)

機械学習モデル: 決定木(sklearn.tree.DecisionTreeClassifier)
主なハイパーパラメータ:

- criterion: gini (default)
- splitter: best (default)
- max_depth: unlimited (default)
- max_leaf_nodes: unlimited (default)

補足:

- パラメータはすべて default に設定する。

この決定木の予測結果は、

- 訓練データに対して accuracy=1.0
- テストデータに対して accuracy=0.9399…

であった。
過学習を起こしているが、テストデータに対しての正解率は約93.99%と高い。

決定木 (チューニング)

機械学習モデル: 決定木(sklearn.tree.DecisionTreeClassifier)
主なハイパーパラメータ:

- criterion: gini (default)
- splitter: best (default)
- max_depth: [6, 9, 12, 15, 18, 21, 24, 27, 30] から最適な値をグリッドサーチで求める
- max_leaf_nodes: unlimited (default)

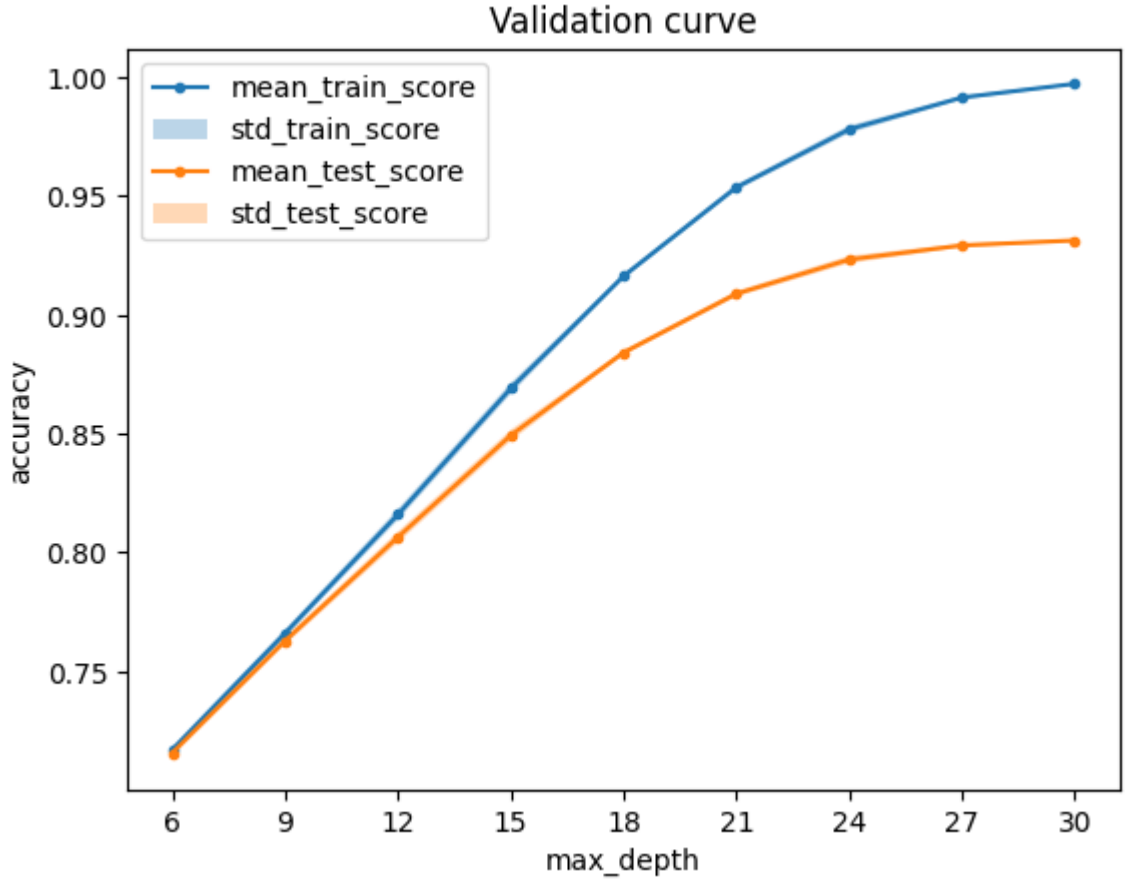
補足:

- max_depth 以外のパラメータはすべて default に設定する。
- グリッドサーチは sklearn.model_selection.GridSearchCV を用いる。探索手法はしらみつぶし探索を採用する。

- 決定木を選んだ理由は、訓練結果から特徴量の重要度を調べることができるためである。詳しくは問2で述べる。

グリッドサーチの結果は以下のようになった。
- Best parameter: max_depth=30
- Best test accuracy: 0.9312…

下図: グリッドサーチにおける検証曲線



この図から次のことが読み取れる.
- accuracy の標準偏差は std_train_score / std_test_score ともに極めて小さい。よって mean_train_score / mean_test_score の値は信用できる。
- max_depth=9 あたりから過学習が始まり、max_depth=27 あたりで過学習は頭打ちになる。

グリッドサーチの結果から、max_depth=30 とすれば良い予測モデルが得られそうだが、過学習を防ぐために最終的なモデルでは max_depth=15 とした。
(予想では、max_depth を深くすればするほど過学習が進み、test accuracy は途中から減少に転じると考えていた。しかし test accuracy が減少することはなかった。この理由として、訓練データと検証データの質が似通っており、モデルが訓練データに適合すればするほど検証データにも適合してしまう、ということが考えられる。あるいは、max_depth をもっと大きくすれば test accuracy が減少するかもしれない)

max_depth=15 の決定木を全訓練データで改めて学習させた結果、
- 訓練データに対する予測結果は accuracy=0.8662…

- テストデータに対する予測結果は accuracy=0.8501…

であった。

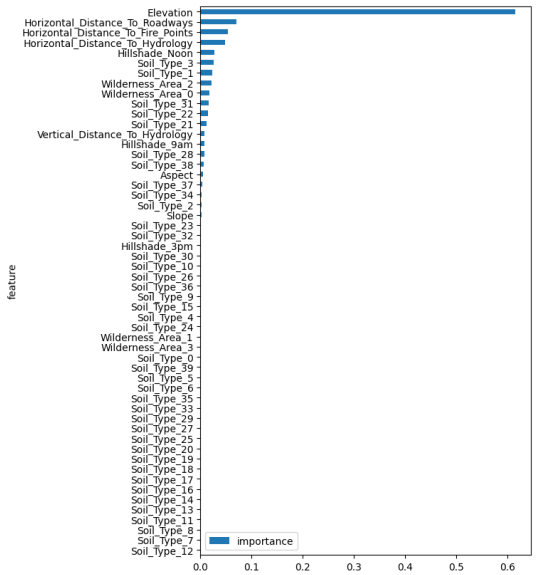
問2

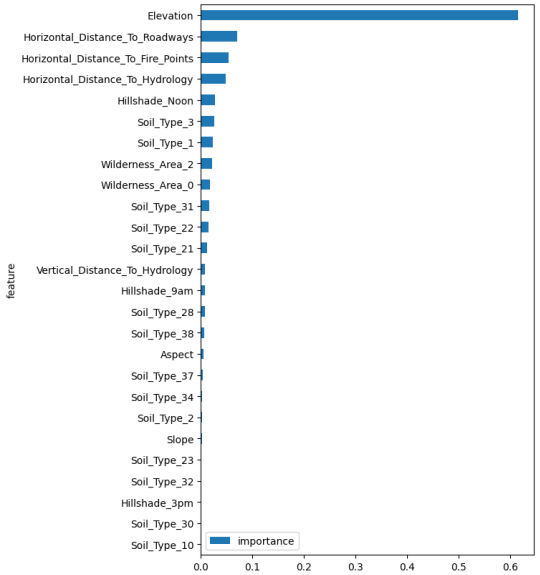
特徴量の比較

問1の決定木から特徴量の重要度 (sklearn.tree.DecisionTreeClassifier.feature_importances_) を算出する。重要度が小さい特徴量を削除し、次元削減を行う。

左下図: 全特徴量の重要度を示したもの
右下図: 左下図から重要度0.001以下の特徴量を削除したもの

以降、次元削減を行ったデータセットを主に使用する。また、次元削減の効果を見るため、元の訓練データで学習させたモデルと新たな訓練データで学習させたモデルを適宜比較する。





機械学習モデルの比較

決定木 (チューニング前)

機械学習モデル: 決定木(sklearn.tree.DecisionTreeClassifier)
主なハイパーパラメータ:

- criterion: gini (default)
- splitter: best (default)
- max_depth: 15

- max_leaf_nodes: unlimited (default)

補足:
- パラメータはすべて default に設定する。
- 訓練データは次元削減したものを用いる。

この決定木の予測結果は、
- 訓練データに対して accuracy=1.0 (問1では0.1)
- テストデータに対して accuracy=0.9384… (問1では0.9399…)

であった。
過学習を起こしているが、テストデータに対しての正解率は約93.84%と高い。

問1の決定木 (元の訓練データで学習) と問2の決定木 (重要度による次元削減を施した訓練データで学習) との間には、正解率の差はあまり見られなかった。後者の方がわずかに正解率が低いこともあり、次元削減の効果は芳しくなさそう。

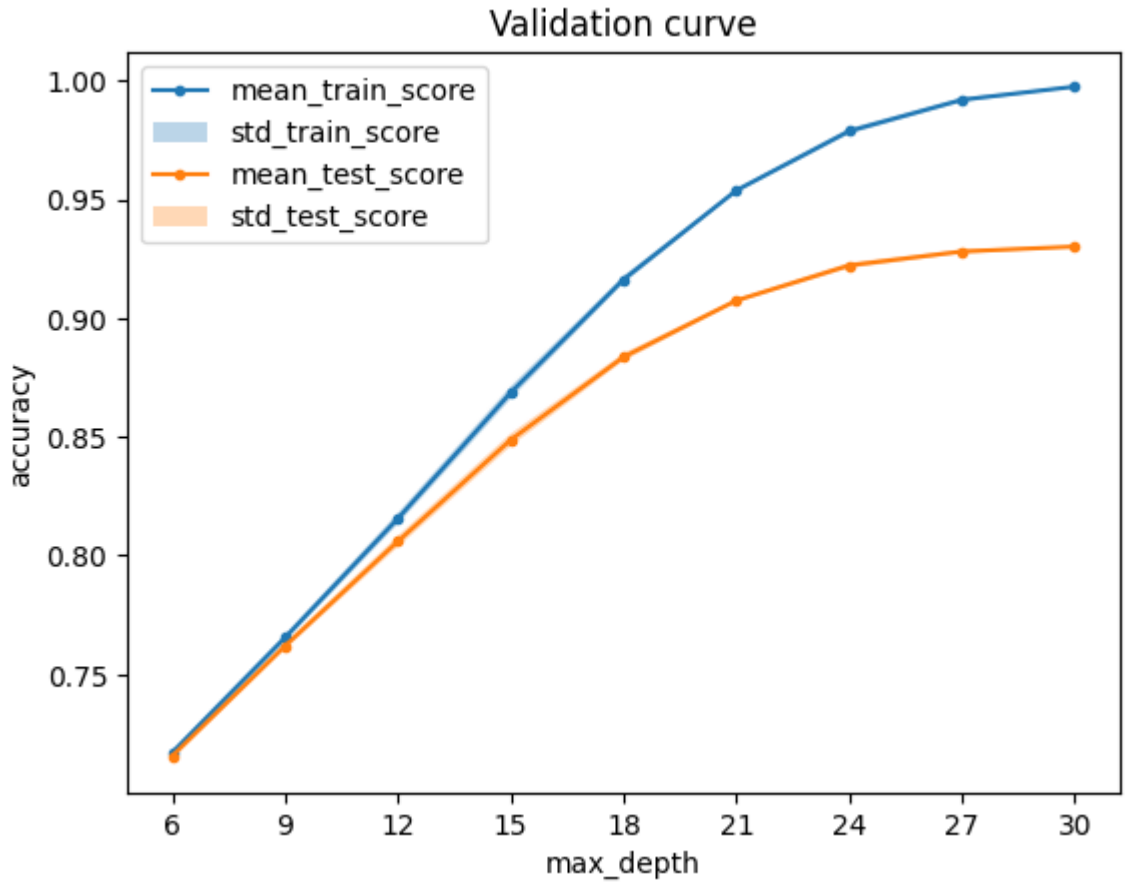
決定木 (チューニング)

機械学習モデル: 決定木(sklearn.tree.DecisionTreeClassifier)
主なハイパーパラメータ:
- criterion: gini (default)
- splitter: best (default)
- max_depth: [6, 9, 12, 15, 18, 21, 24, 27, 30] から最適な値をグリッドサーチで求める
- max_leaf_nodes: unlimited (default)

補足:
- max_depth 以外のパラメータはすべて default に設定する。
- グリッドサーチは sklearn.model_selection.GridSearchCV を用いる。探索手法はしらみつぶし探索を採用する。
- 訓練データは次元削減したものを用いる。

グリッドサーチの結果は以下のようになった。
- Best parameter: max_depth=30
- Best test accuracy: 0.9301…

下図: グリッドサーチにおける検証曲線



問1の検証曲線 (元の訓練データで学習) と問2の検証曲線 (重要度による次元削減を施した訓練データで学習) はほとんど同じであった。

max_depth=15 の決定木を次元削減済みの全訓練データを用いて改めて学習させた結果、
- 訓練データに対する予測結果は accuracy=0.8670… (問1では0.8662…)
- テストデータに対する予測結果は accuracy=0.8500… (問1では0.8501…)

であった。

問1のチューニングした決定木 (元の訓練データで学習) と問2のチューニングした決定木 (重要度による次元削減を施した訓練データで学習) との間には、性能の差はあまり見られなかった。

ランダムフォレスト (チューニング前)

元の訓練データを用いた場合

機械学習モデル: ランダムフォレスト(sklearn.ensemble.RandomForestClassifier)
主なハイパーパラメータ:
- n_estimators: 100 (default)
- max_depth: unlimited (default)
- bootstrap: True (default)

補足:
- パラメータはすべて default に設定する。
- 訓練データは元のものを用いる。

このランダムフォレストの予測結果は、
- 訓練データに対して accuracy=1.0
- テストデータに対して accuracy=0.9548…

であった。

次元削減済みの訓練データを用いた場合

機械学習モデル: ランダムフォレスト(sklearn.ensemble.RandomForestClassifier)
主なハイパーパラメータ:
- n_estimators: 100 (default)
- max_depth: unlimited (default)
- bootstrap: True (default)

補足:
- パラメータはすべて default に設定する。
- 訓練データは次元削減したものを用いる。

このランダムフォレストの予測結果は、
- 訓練データに対して accuracy=1.0
- テストデータに対して accuracy=9.9563…

であった。

以上二つのランダムフォレストを比較すると、正解率の差はあまり見られなかった。後者の方がわずかに正解率が高いが、次元削減の効果が芳しいとは言えない。

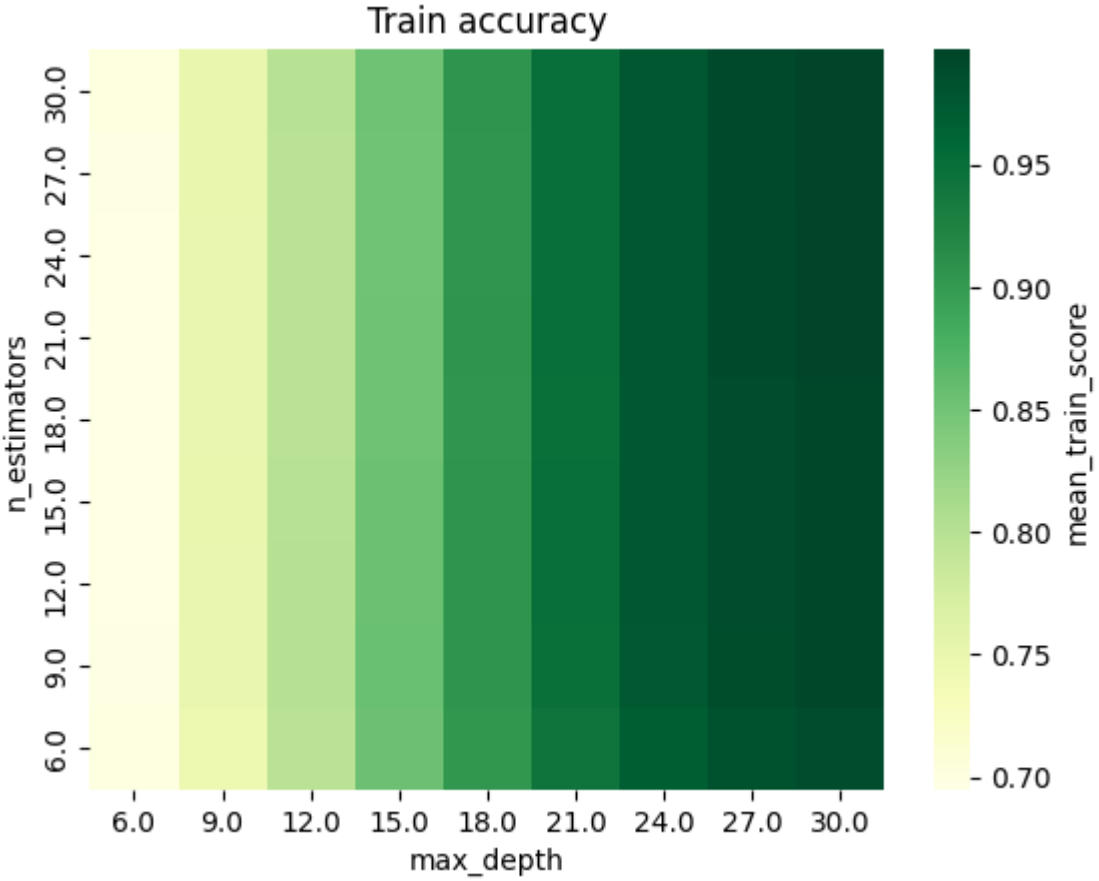
ランダムフォレスト (チューニング)

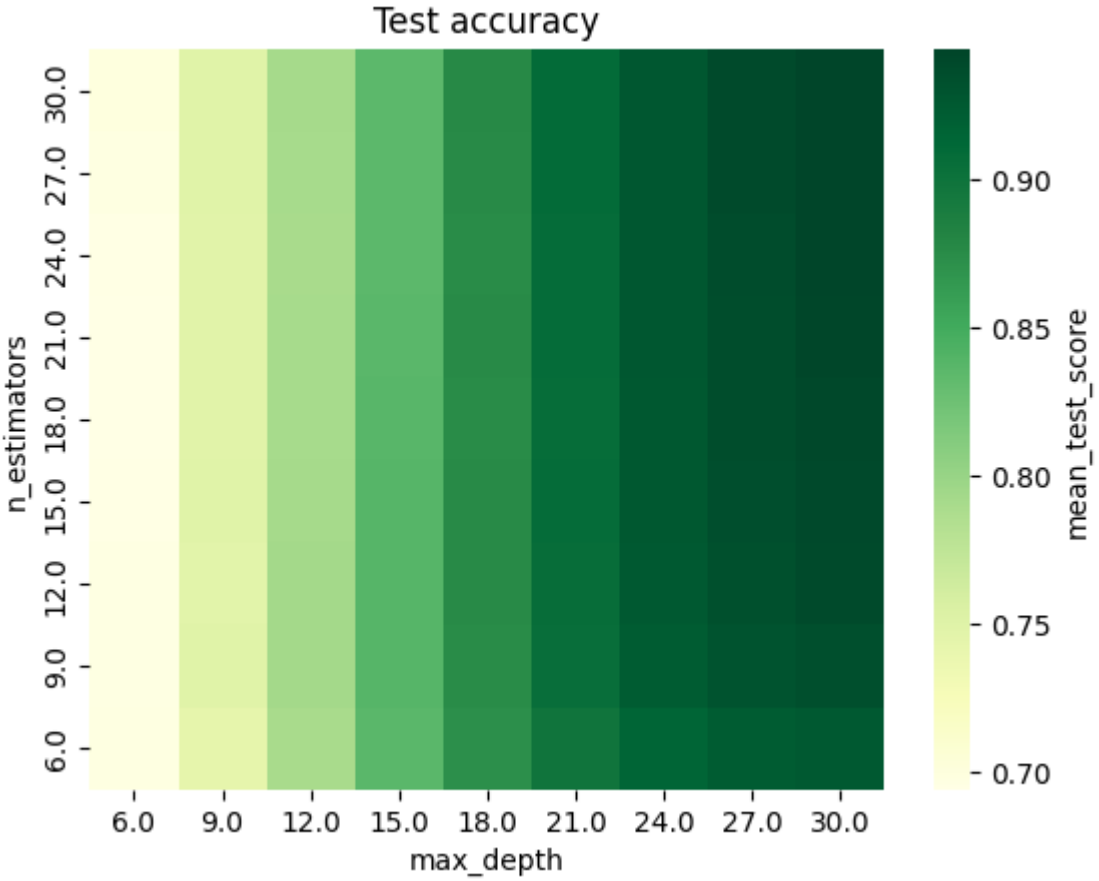
機械学習モデル: ランダムフォレスト(sklearn.ensemble.RandomForestClassifier)
主なハイパーパラメータ:
- n_estimators: [6, 9, 12, 15, 18, 21, 24, 27, 30] から最適な値をグリッドサーチで求める
- max_depth: [6, 9, 12, 15, 18, 21, 24, 27, 30] から最適な値をグリッドサーチで求める
- bootstrap: True (default)

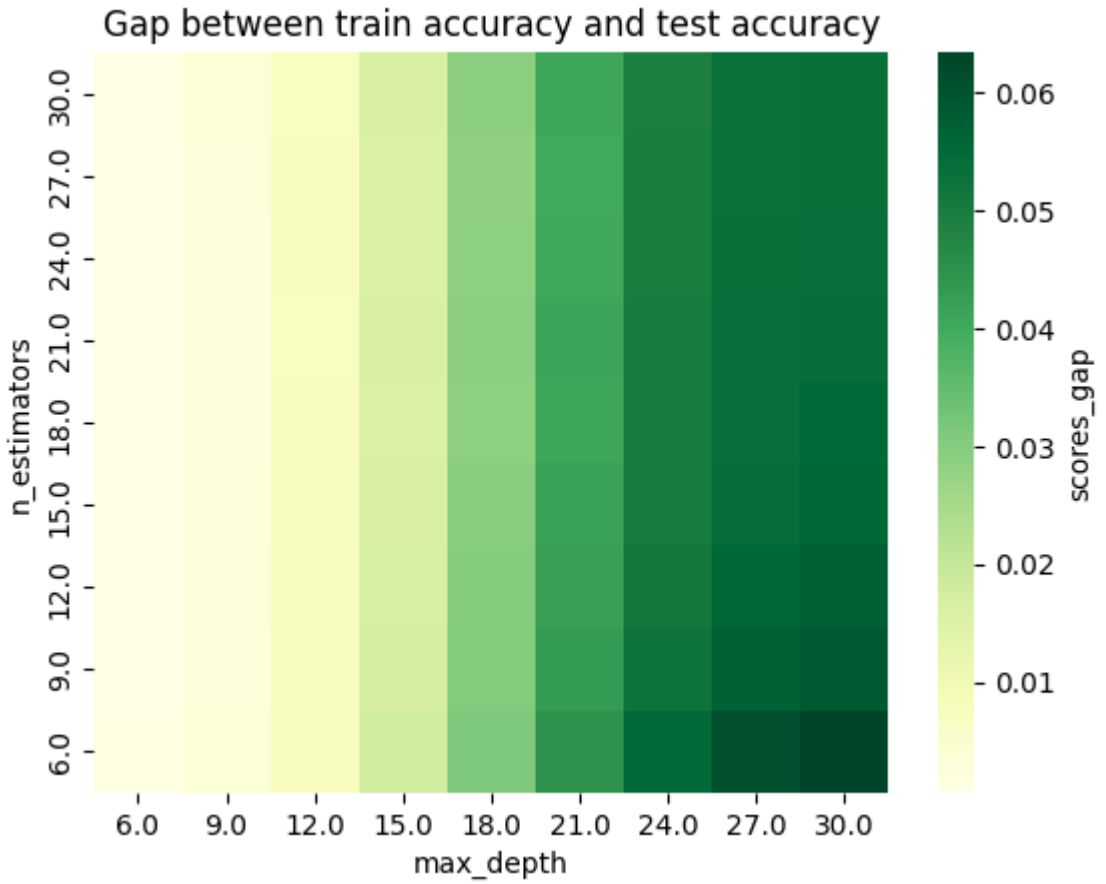
補足:
- n_estimators と max_depth 以外のパラメータはすべて default に設定する。
- 訓練データは次元削減したものを用いる。

グリッドサーチの結果は以下のようになった。
- Best parameter: max_depth=30, n_estimators=30
- Best test accuracy: 0.9438…

下図: グリッドサーチにおける検証曲線を表すヒートマップ
- 1枚目: 訓練データに対する accuracy
- 2枚目: 検証データに対する accuracy
- 3枚目: 上記二つの accuracy の差 (差が大きいほど過学習が進んでいる)


Train accuracy
n_estimators
6.0
9.0
12.0
15.0
18.0
21.0
24.0
27.0
30.0
max_depth
mean_train_score
0.95
0.90
0.85
0.80
0.75
0.70


Test accuracy
n_estimators
6.0
9.0
12.0
15.0
18.0
21.0
24.0
27.0
30.0
max_depth
mean_test_score
0.90
0.85
0.80
0.75
0.70



これらの図から次のことが読み取れる.

- n_estimators の値が大きいほど、ほんのわずかに accuracy が高くなる。
- max_depth の値が大きいほど、明らかに accuracy が高くなる。
- n_estimators の値が小さいほど、わずかに過学習が進む。
- max_depth の値が大きいほど、明らかに過学習が進む。

よって、n_estimators の値は sklearn のデフォルトの100に設定し、max_depth の値は決定木の場合と同様にして15と設定した。このランダムフォレストを次元削減済みの全訓練データを用いて改めて学習させた結果、

- 訓練データに対する予測結果は accuracy=0.8556… (0.8423…)
- テストデータに対する予測結果は accuracy=0.8409… (0.8288…)

であった。かっこ内の数値は、元の訓練データで学習させたランダムフォレストの予測結果。

かっこ内の数値と比較すると、次元削減済みの訓練データで学習させたランダムフォレストの方が正解率が1.5%ほど高い。よって次元削減の効果はあったと考えられる。

一方で、問2の決定木と比較すると、

- 決定木: accuracy=0.8500…
- ランダムフォレスト: accuracy=0.8409…

と、ランダムフォレストの方が正解率がわずかに劣る。

まとめ

決定木の max_depth の値が大きいほど正解率が高いことや、ランダムフォレストよりも決定木の方が正解率が高いことから、このデータセットにおいては過学習が進むほどテストデータの正解率が高くなると結論づけられる。
もし今回のデータセットが未知のデータをうまく表現しているのであれば、max_depth の値を大きくしてもいいかもしれない。しかし、地球上の forest cover を完璧に表現しているとは思われない。
よって、最終的なモデルはより高い汎化性能が期待できる n_estimators=100, max_depth=15 のランダムフォレストを採用する。