リンクテキスト# 学籍番号 C0B23107 氏名 仲田祥吾

データサイエンス 第11回課題 決定木生成とグリッドサーチ

各問題のソースと実行結果をPDF形式で提出しなさい

- 冒頭の**学籍番号・氏名**を正しく入力してください
- 提出ファイル名は、「**11_決定木_分類木_GS_学籍番号_氏名.pdf**」とする。

課題11-1

- 決定木アルゴリズムを用いてアヤメデータの3つの品種（virginica、setosa、versicolor）を分類するための予測モデルを生成し、予測精度を評価せよ。
- ただし、「木の深さ(の最大値)」のハイパーパラメータを交差検証（クロスバリデーション）を用いたグリッドサーチにより探索せよ。交差検証では学習データを10分割することとし、木の深さの最大値を1から5で試せ。
- 決定木の変数重要度も出力せよ

必要に応じてセルを増やしなさい

In [3]:
```
# Googleドライブをマウントしてデータの読み込み・保存に利用する
from google.colab import drive
drive.mount('/content/drive')
%cd "/content/drive/MyDrive/Colab Notebooks/DS2024H2"
# グラフへの日本語対応
! pip install japanize_matplotlib
import matplotlib.pyplot as plt
import japanize_matplotlib
import pandas as pd
df_iris = pd.read_csv("iris.csv", encoding='utf_8_sig')
display(df_iris)
df_iris['target'] = df_iris['species'].map({'setosa': 0, 'virginica': 1, 'versic
display(df_iris[0:5],df_iris[50:55],df_iris[100:105])
# 説明変数のデータを抽出
X = df_iris[["sepal_length","sepal_width","petal_length","petal_width"]]
# 目的変数のデータを抽出
y = df_iris['target']
from sklearn.model_selection import train_test_split
# 学習データ7割、評価データ３割に分割
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,stratify
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
# 木の深さ（の最大値）を1から5まで変化させるように設定
param = {'max_depth':[1, 2, 3, 4, 5]}
# 学習データを5つに分割した交差検証（クロスバリデーション）
clf_gs = GridSearchCV(DecisionTreeClassifier(random_state=0), # 決定木を使用（乱数
                      param, # （探索するハイパーパラメータ）
                      cv=10, # （交差検証の分割数）
```

```
                        scoring='accuracy').fit(X_train, y_train)
# スコアとパラメータの組合せ
params = clf_gs.cv_results_['params']
scores = clf_gs.cv_results_['mean_test_score']
for i in range(len(scores)):
  print(params[i], "の平均スコア(正解率)", scores[i].round(10)) #小数3桁表示
#最良のモデル抽出
best_clf = clf_gs.best_estimator_

print('最良条件(木の深さの最大値)：', best_clf.max_depth)

print('学習データに対するスコア(正解率):', round(best_clf.score(X_train, y_train),
print('評価データに対するスコア(正解率):', round(best_clf.score(X_test, y_test), 3)
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier, plot_tree
cols_names  = X_train.columns # 説明変数の列名ラベル
class_names = ['setosa','versicolor','virginica'] # 品種ラベル（0: setosa, 1: ver
plt.figure(figsize=(7, 7))
plot_tree(best_clf, feature_names=cols_names, class_names=class_names, filled=Tr
plt.show()
df_importances = pd.DataFrame() # 空のデータフレーム作成
df_importances["変数"]   = X.columns # 変数名のリストをデータフレームに追加
df_importances["重要度"] = best_clf.feature_importances_ # 変数重要度をデータフレー
display(df_importances.round(3)) # 小数3桁で表示
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call dri
ve.mount("/content/drive", force_remount=True).
/content/drive/MyDrive/Colab Notebooks/DS2024H2
Requirement already satisfied: japanize_matplotlib in /usr/local/lib/python3.10/d
ist-packages (1.1.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packa
ges (from japanize_matplotlib) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist
-packages (from matplotlib->japanize_matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-pac
kages (from matplotlib->japanize_matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dis
t-packages (from matplotlib->japanize_matplotlib) (4.55.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dis
t-packages (from matplotlib->japanize_matplotlib) (1.4.7)
Requirement already satisfied: numpy<2,>=1.21 in /usr/local/lib/python3.10/dist-p
ackages (from matplotlib->japanize_matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->japanize_matplotlib) (24.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-pa
ckages (from matplotlib->japanize_matplotlib) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist
-packages (from matplotlib->japanize_matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/
dist-packages (from matplotlib->japanize_matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-package
s (from python-dateutil>=2.7->matplotlib->japanize_matplotlib) (1.16.0)
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

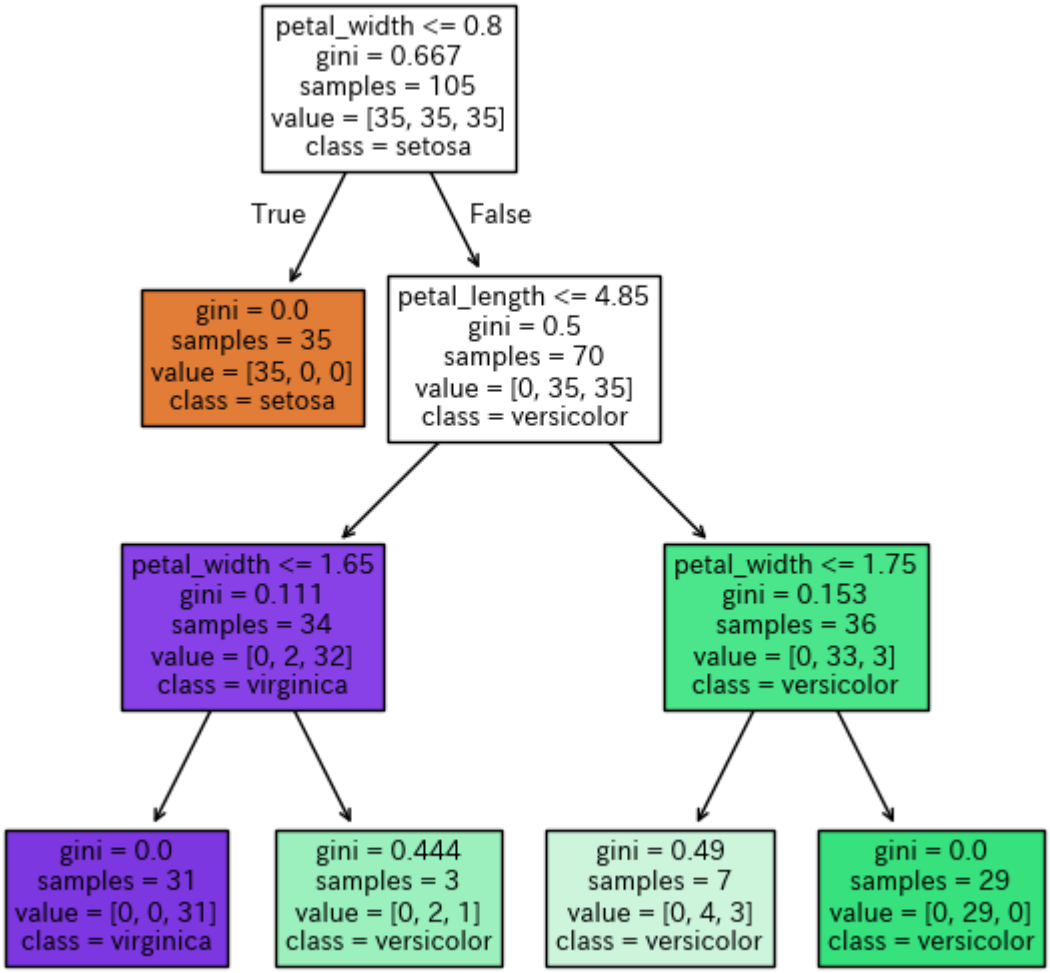
150 rows × 5 columns

	sepal_length	sepal_width	petal_length	petal_width	species	target
0	5.1	3.5	1.4	0.2	setosa	0
1	4.9	3.0	1.4	0.2	setosa	0
2	4.7	3.2	1.3	0.2	setosa	0
3	4.6	3.1	1.5	0.2	setosa	0
4	5.0	3.6	1.4	0.2	setosa	0

	sepal_length	sepal_width	petal_length	petal_width	species	target
50	7.0	3.2	4.7	1.4	versicolor	2
51	6.4	3.2	4.5	1.5	versicolor	2
52	6.9	3.1	4.9	1.5	versicolor	2
53	5.5	2.3	4.0	1.3	versicolor	2
54	6.5	2.8	4.6	1.5	versicolor	2

	sepal_length	sepal_width	petal_length	petal_width	species	target
100	6.3	3.3	6.0	2.5	virginica	1
101	5.8	2.7	5.1	1.9	virginica	1
102	7.1	3.0	5.9	2.1	virginica	1
103	6.3	2.9	5.6	1.8	virginica	1
104	6.5	3.0	5.8	2.2	virginica	1

```
{'max_depth': 1} の平均スコア(正解率) 0.6681818182
{'max_depth': 2} の平均スコア(正解率) 0.9136363636
{'max_depth': 3} の平均スコア(正解率) 0.9327272727
{'max_depth': 4} の平均スコア(正解率) 0.9045454545
{'max_depth': 5} の平均スコア(正解率) 0.9045454545
最良条件(木の深さの最大値)： 3
学習データに対するスコア(正解率): 0.962
評価データに対するスコア(正解率): 0.978
```



	変数	重要度
0	sepal_length	0.000
1	sepal_width	0.000
2	petal_length	0.394
3	petal_width	0.606

結果のまとめをこちらに記載 petal_widthのみで決定木が完成しているので、変数重要度はpetal_width=1.0で他は0となる。
