

```
# さまざまな depth に対して繰り返す
for depth in range(1, 25):
    # モデルの初期化
    clf = tree.DecisionTreeClassifier(max_depth=depth)
    # 学習に利用する特徴量を指定
    # この部分はループの外で実行可能
    cols = [
        'fixed acidity',
        'volatile acidity',
        'citric acid',
        'residual sugar',
        'chlorides',
        'free sulfur dioxide',
        'total sulfur dioxide',
        'density',
        'pH',
        'sulphates',
        'alcohol'
    ]
    # 与えられた特徴量と対応する目的変数でモデルを学習
    clf.fit(df_train[cols], df_train.quality)

    # 学習用と検証用データセットに対する予測
    train_predictions = clf.predict(df_train[cols])
    test_predictions = clf.predict(df_test[cols])
    # 学習用と検証用データセットに対する正答率を計算
    train_accuracy = metrics.accuracy_score(
        df_train.quality, train_predictions
    )
    test_accuracy = metrics.accuracy_score(
        df_test.quality, test_predictions
    )
    # リストに計算結果を追加
    train_accuracies.append(train_accuracy)
    test_accuracies.append(test_accuracy)

# matplotlib と seaborn による可視化
plt.figure(figsize=(10, 5))
sns.set_style("whitegrid")
plt.plot(train_accuracies, label="train accuracy")
plt.plot(test_accuracies, label="test accuracy")
plt.legend(loc="upper left", prop={'size': 15})
plt.xticks(range(0, 26, 5))
plt.xlabel("max_depth", size=20)
plt.ylabel("accuracy", size=20)
plt.show()
```