【データサイエンス［J23］レポート課題】

1.正規化と標準化

佐々木優人 一関高専　情報・ソフトウェア系　4年17番f20075 2023/10/09

1. 作成したプログラム

1.1 正規化

正規化とは、データの最小値からの偏差（＝最小値を中心0にした場合の値）をデータ範囲（＝最大値－最小値）で割ることである。これにより、データの最小値は0、最大値は1に変換される[1]。これをscikit-learnの実装規約を参考にして実装したプログラムを以下に示す。

MinMaxscaler.py

```
class MinMaxScaler(OneToOneFeatureMixin, TransformerMixin, BaseEstimator):
    def __init__(self, in_features=None):
        self.in_features = in_features

    def fit(self, X, y=None):
        if self.in_features is None:
            self.in_features_ = np.ones(X.shape[1], dtype=np.bool_)
        else:
            self.in_features_ = self.in_features
        X = X[:,self.in_features_]
        self.min_ = X.min(axis=0)
        self.max_ = X.max(axis=0)
        return self

    def transform(self, X):
        check_is_fitted(self, "min_")
        X = X[:,self.in_features_]
        transformed = (X - self.min_) / (self.max_ - self.min_)
        return transformed

    def fit_transform(self, X):
        self.fit(X)
        tranformed = self.transform(X)
        return tranformed
```

1.2 標準化

標準化とは、データの平均値からの偏差（＝平均値を中心0にした場合の値、中心化した値）を標準偏差で割ることである。これにより、データの平均は0、分散（標準偏差）は1に変換される。これをscikit-learnの実装規約を参考にして実装したプログラムを以下に示す。

StandardScaler.py

```
class StandardScaler(OneToOneFeatureMixin, TransformerMixin,
BaseEstimator):
    def __init__(self, in_features: Sequence[int] | None =None,ddof=0)-
>None:
        self.in_features = in_features
        self.ddof=ddof

    def fit(self, X: Num[Array, "data feature"], y=None)->Self:
        if self.in_features is None:
            self.in_features_ = np.ones(X.shape[1], dtype=np.bool_)
        else:
            self.in_features_ = self.in_features
        X = X[:,self.in_features_]
        self.mu_ = X.mean(axis=0)
        self.sigma = X.std(ddof=self.ddof,axis=0)
        return self

    def transform(self, X: Num[Array, "data feature"])->Num[Array, "data
feature"]:
        check_is_fitted(self, "mu_")
        X = X[:,self.in_features_]
        transformed = (X - self.mu_) / self.sigma
        return transformed

    def fit_transform(self, X: Num[Array, "data feature"])->Num[Array,
"data feature"]:
        self.fit(X)
        tranformed = self.transform(X)
        return tranformed
```

2. 実行結果

上記の二つのプログラムの実行結果を以下に示す。比較のためにこれらの処理を適用する前の教師データの箱ひげ図も示す[2]。

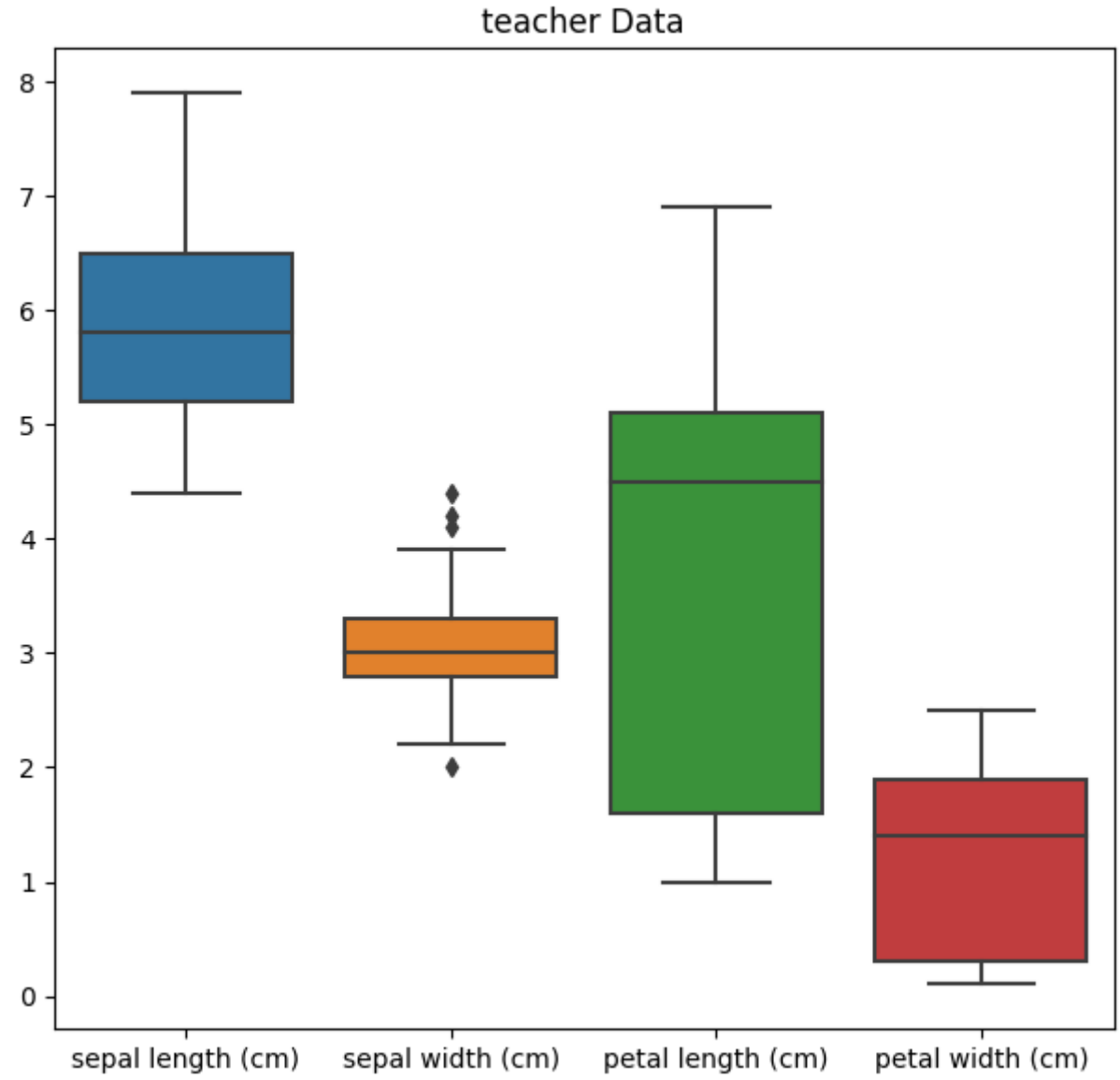
teacher Data
8
7
6
5
4
3
2
1
0
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

図1：生データ

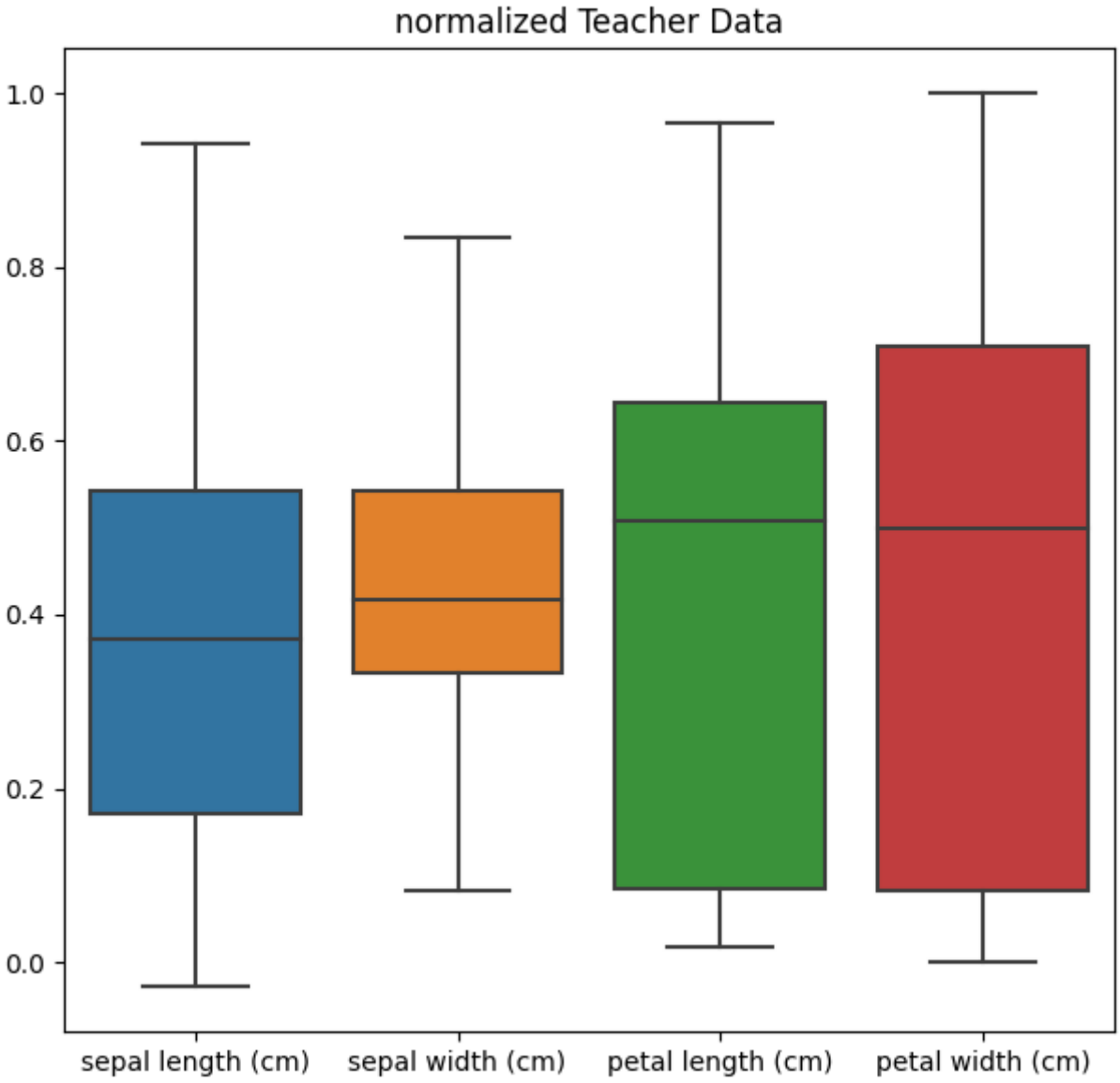
normalized Teacher Data
1.0
0.8
0.6
0.4
0.2
0.0
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

図２: 正規化

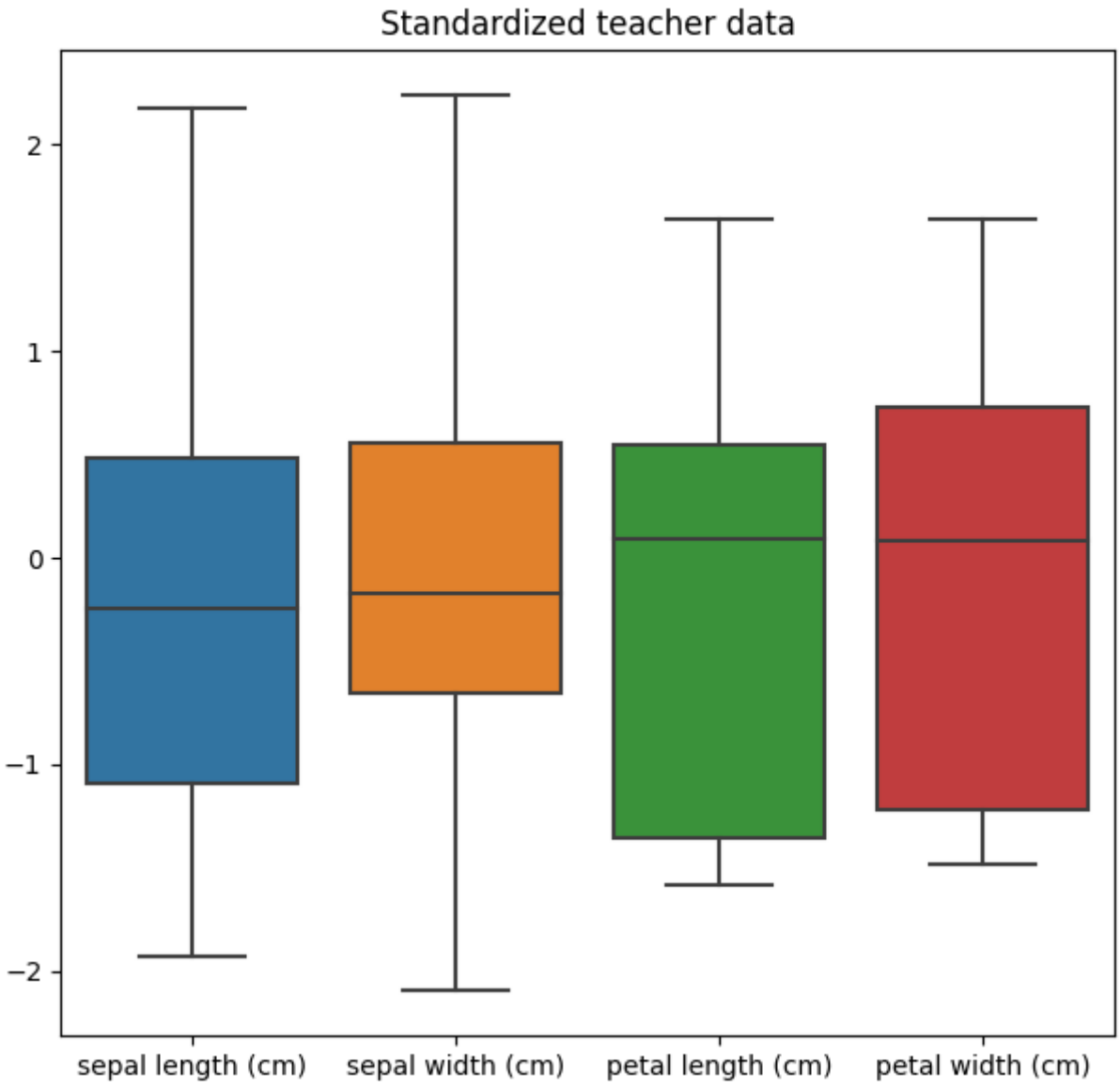

図３: 標準化

以上の結果から、正規化を行った結果、petal lengthとpetal widthは、speal lengthとsepal widthと比べて、第３四分位点と第１四分位点の差が大きいことがわかる。標準化を行った結果、全ての要素の第３四分位点と第１四分位点の差は少ないことがわかる。

３. 正規化と標準化の利点と欠点

3.1 正規化

3.1.1 利点

最大値や最小値が決まっている場合、正規化を使うことで学習コストが低くなる利点がある。画像処理の場合、ピクセルのRGB値など、最大値と最小値が明確に決まっているものに使われることが多い。

3.1.2 欠点

最大値や最小値が決まっていない場合、外れ値に大きく影響されてしまうため、特徴がうまく現れない欠点がある。

3.2 標準化

3.2.1 利点

標準化は、データの平均と標準偏差を用いて行うため、外れ値に影響を受けにくい利点がある。

3.2.2 欠点

データが正規分布していない場合は、適さないという欠点がある。

4. 統計値をテストデータから得た場合の問題点

機械学習モデルを作る際に、未知のデータに対しての性能を測ることで適切に評価することができるが、テストデータを用いて学習してしまうと、そのテストデータだけに対したモデルが出来上がる問題点がある。

参考文献

[1]一色政彦,正規化（Normalization）／標準化（Standardization）とは？,https://atmarkit.itmedia.co.jp/ait/articles/2110/07/news027.html, 訪問日 2023/10/09

[2]リファレンスメモ,Python-matplotlib : boxplot【箱ひげ図 (データ指定)】・bxp【箱ひげ図 (要約統計量指定)】,https://cercopes-z.com/Python/matplotlib/graph-boxplot-mpl.html, 訪問日 2023/10/09