

Рубежный контроль

Хуан Цзэсян

Вариант:20

Задача1:20. Задача2:40.

Задача1:20

Для набора данных проведите масштабирование данных для одного (произвольного) числового признака с использованием MinMax-масштабирования. для произвольной колонки данных построить график "Скрипичная диаграмма (violin plot)".

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_diabetes
from sklearn.preprocessing import MinMaxScaler

# Загрузка набора данных по диабету
diabetes = load_diabetes()
data = diabetes.data
feature_names = diabetes.feature_names
target = diabetes.target

# Создание DataFrame
df = pd.DataFrame(data, columns=feature_names)
df['target'] = target

# Вывод первых десяти строк исходных данных
print("Исходные данные (первые десять строк):")
display(df.head(10)) # Использование функции display для
отображения в виде таблицы

# Выбор числового признака для масштабирования MinMax (например,
столбец 'bmi')
scaler = MinMaxScaler()
df['bmi_scaled'] = scaler.fit_transform(df[['bmi']])

# Вывод первых десяти строк масштабированных данных
print("\nМасштабированные данные (первые десять строк):")
display(df[['bmi', 'bmi_scaled']].head(10)) # Использование
функции display для отображения в виде таблицы
```

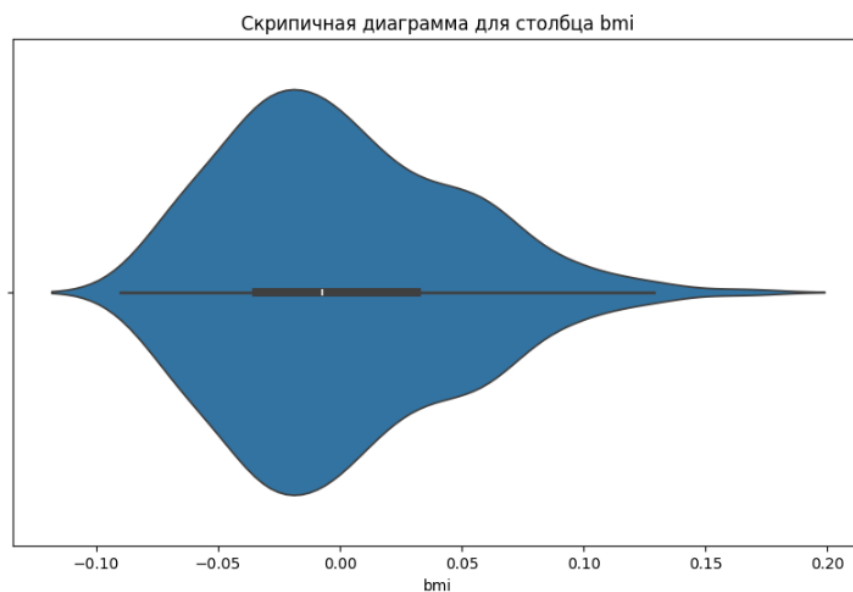
```
# Построение скрипичной диаграммы для произвольного столбца
(например, столбец 'bmi')
plt.figure(figsize=(10, 6))
sns.violinplot(x=df['bmi'])
plt.title('Скрипичная диаграмма для столбца bmi')
plt.xlabel('bmi')
plt.show()
```

Исходные данные (первые десять строк):

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641	135.0
5	-0.092695	-0.044642	-0.040696	-0.019442	-0.068991	-0.079288	0.041277	-0.076395	-0.041176	-0.096346	97.0
6	-0.045472	0.050680	-0.047163	-0.015999	-0.040096	-0.024800	0.000779	-0.039493	-0.062917	-0.038357	138.0
7	0.063504	0.050680	-0.001895	0.066629	0.090620	0.108914	0.022869	0.017703	-0.035816	0.003064	63.0
8	0.041708	0.050680	0.061696	-0.040099	-0.013953	0.006202	-0.028674	-0.002592	-0.014960	0.011349	110.0
9	-0.070900	-0.044642	0.039062	-0.033213	-0.012577	-0.034508	-0.024993	-0.002592	0.067737	-0.013504	310.0

Масштабированные данные (первые десять строк):

	bmi	bmi_scaled
0	0.061696	0.582645
1	-0.051474	0.148760
2	0.044451	0.516529
3	-0.011595	0.301653
4	-0.036385	0.206612
5	-0.040696	0.190083
6	-0.047163	0.165289
7	-0.001895	0.338843
8	0.061696	0.582645
9	0.039062	0.495868



Задача2:40

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте метод на основе корреляции признаков. для произвольной колонки данных построить график "Скрипичная диаграмма (violin plot)".
Выбор функций на основе корреляции между функциями и целевыми переменными

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.preprocessing import MinMaxScaler
import numpy as np

# Загрузка данных об ирисах
iris = load_iris()
data = iris.data
feature_names = iris.feature_names
target = iris.target

# Создание DataFrame
df = pd.DataFrame(data, columns=feature_names)
df['target'] = target

# Вывод первых десяти строк данных (первые десять строк)
print("Исходные данные (первые десять строк):")
display(df.head(10))

# Вычисление матрицы корреляций
correlation_matrix = df.corr().abs()

# Выбор признаков с наибольшей корреляцией с целевой переменной
target_corr =
correlation_matrix['target'].sort_values(ascending=False)

# Выбор двух признаков с наибольшей корреляцией с целевой
переменной
selected_features = target_corr.index[1:3].tolist()
print("\nВыбранные признаки:", selected_features)

# Масштабирование выбранных признаков (например, признак 'petal
width (cm)')
scaler = MinMaxScaler()
```

```

df['petal width (cm)_scaled'] = scaler.fit_transform(df[['petal
width (cm)']])

# Вывод масштабированных данных (первые десять строк)
print("\nМасштабированные данные (первые десять строк):")
display(df[['petal width (cm)', 'petal width
(cm)_scaled']].head(10))

# Построение скрипичной диаграммы для выбранного признака
(например, 'petal width (cm)')
plt.figure(figsize=(10, 6))
sns.violinplot(x=df['petal width (cm)'])
plt.title('Скрипичная диаграмма для признака petal width (cm)')
plt.xlabel('petal width (cm)')
plt.show()

```

Исходные данные (первые десять строк):

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0
7	5.0	3.4	1.5	0.2	0
8	4.4	2.9	1.4	0.2	0
9	4.9	3.1	1.5	0.1	0

Выбранные признаки: ['petal width (cm)', 'petal length (cm)']

Масштабированные данные (первые десять строк):

	petal width (cm)	petal width (cm)_scaled
0	0.2	0.041667
1	0.2	0.041667
2	0.2	0.041667
3	0.2	0.041667
4	0.2	0.041667
5	0.4	0.125000
6	0.3	0.083333
7	0.2	0.041667
8	0.2	0.041667
9	0.1	0.000000

Скрипичная диаграмма для признака petal width (cm)

