Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления и искусственный интеллект

КАФЕДРА Системы обработки информации и управления

Рубежный контроль №1 по курсу «Методы машинного обучения в автоматизированных системах обработки информации и управления»

Подготовили:

У Жун

ИУ5И-25М

12.04.2024

Проверил:

Гапанюк Ю. Е.

2024 г.

Варианты заданий

Номер варианта	Номер задачи №1	Номер задачи №2	дополнительная задача
2 +15=17	17	37	построить парные диаграммы (pairplot)

Задача №17.

Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием преобразования Йео-Джонсона (Yeo-Johnson transformation).

Для загрузки набора данных по диабету была использована библиотека scikit-learn, а в качестве имени признака было выбрано "bmi" (индекс массы тела).

[1]:
```
import numpy as np
import pandas as pd
from sklearn.datasets import load_diabetes
from sklearn.preprocessing import PowerTransformer
# 加载糖尿病数据集
diabetes_data = load_diabetes()
# 创建 DataFrame，使用数据和特征名称
df = pd.DataFrame(data=diabetes_data.data, columns=diabetes_data.feature_names)
# 选择要归一化的特征名称
feature_name = 'bmi'
# 初始化 Yeo-Johnson 转换器
yeo_johnson_transformer = PowerTransformer(method='yeo-johnson')

# 对选定特征进行归一化
df_transformed = df.copy()  # 复制 DataFrame 以保留原始数据
df_transformed[feature_name] = yeo_johnson_transformer.fit_transform(df[[feature_name]])
# 打印转换后的结果
print(df_transformed.head())
```

```
        age       sex       bmi        bp        s1        s2        s3  \
0  0.038076  0.050680  1.272058  0.021872 -0.044223 -0.034821 -0.043401
1 -0.001882 -0.044642 -1.159100 -0.026328 -0.008449 -0.019163  0.074412
2  0.085299  0.050680  0.984937 -0.005670 -0.045599 -0.034194 -0.032356
3 -0.089063 -0.044642 -0.139725 -0.036656  0.012191  0.024991 -0.036038
4  0.005383 -0.044642 -0.749303  0.021872  0.003935  0.015596  0.008142

         s4        s5        s6
0 -0.002592  0.019907 -0.017646
1 -0.039493 -0.068332 -0.092204
2 -0.002592  0.002861 -0.025930
3  0.034309  0.022688 -0.009362
```

Просмотр сводной статистики для преобразованных функций:

[4]:
```
# 查看转换后特征的统计摘要
print(df_transformed[[feature_name]].describe())
```

```
                bmi
count  4.420000e+02
mean  -2.411344e-17
std    1.001133e+00
min   -2.362399e+00
25%   -6.932270e-01
50%   -4.131995e-02
75%    7.479512e-01
max    2.630623e+00
```

Задача №39

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectPercentile для 5% лучших признаков, и метод, основанный на взаимной информации.

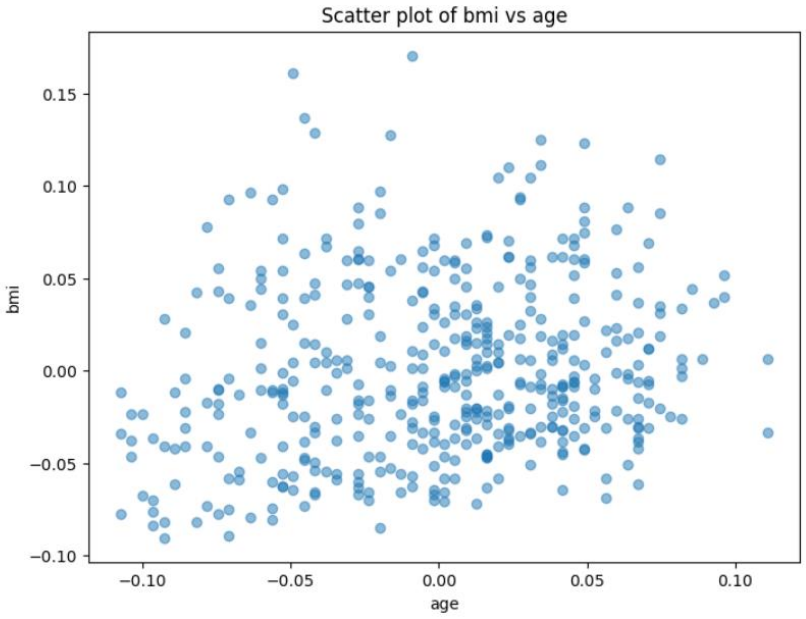
[12]:
```
#37
import numpy as np
import pandas as pd
from sklearn.datasets import load_diabetes
from sklearn.feature_selection import SelectPercentile, mutual_info_regression
np.random.seed(0)  # 设置随机种子

# 加载糖尿病数据集
diabetes_data = load_diabetes()
X = diabetes_data.data  # 特征矩阵
y = diabetes_data.target  # 目标变量
feature_names = diabetes_data.feature_names  # 特征名称
# 计算特征与目标变量之间的互信息
mutual_info = mutual_info_regression(X, y)
# 选择 5% 最佳特征
percentile = 5
num_features = int(len(feature_names) * percentile / 100)

selector = SelectPercentile(mutual_info_regression, percentile=percentile)
selector.fit(X, y)

# 获取选定特征的索引
selected_features_indices = selector.get_support(indices=True)

# 获取选定特征的名称
selected_features = [feature_names[i] for i in selected_features_indices]
# 输出特征选择结果
print(f"Выбранные признаки ({percentile}% лучших):")
for feature, mi_score in zip(selected_features, selector.scores_[selected_features_indices]):
    print(f"{feature}: {mi_score:.3f}")
```

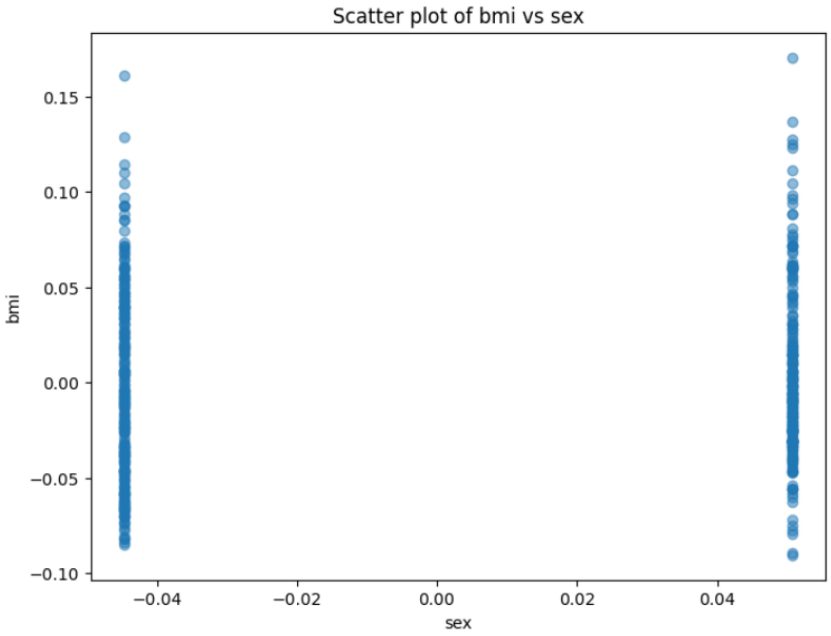
```
Выбранные признаки (5% лучших):
bmi: 0.175
```

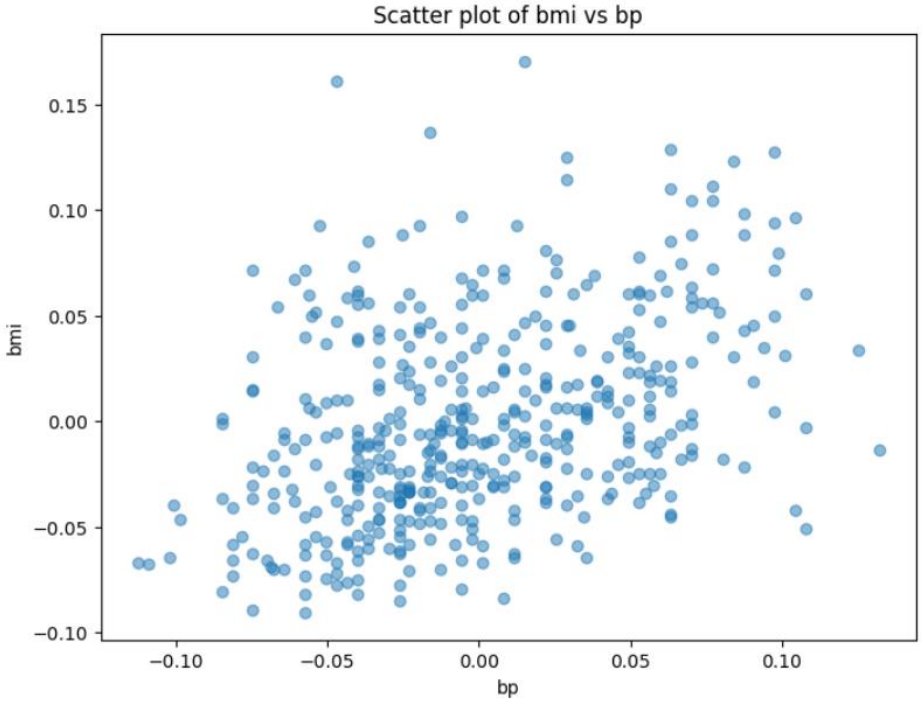
дополнительная задача

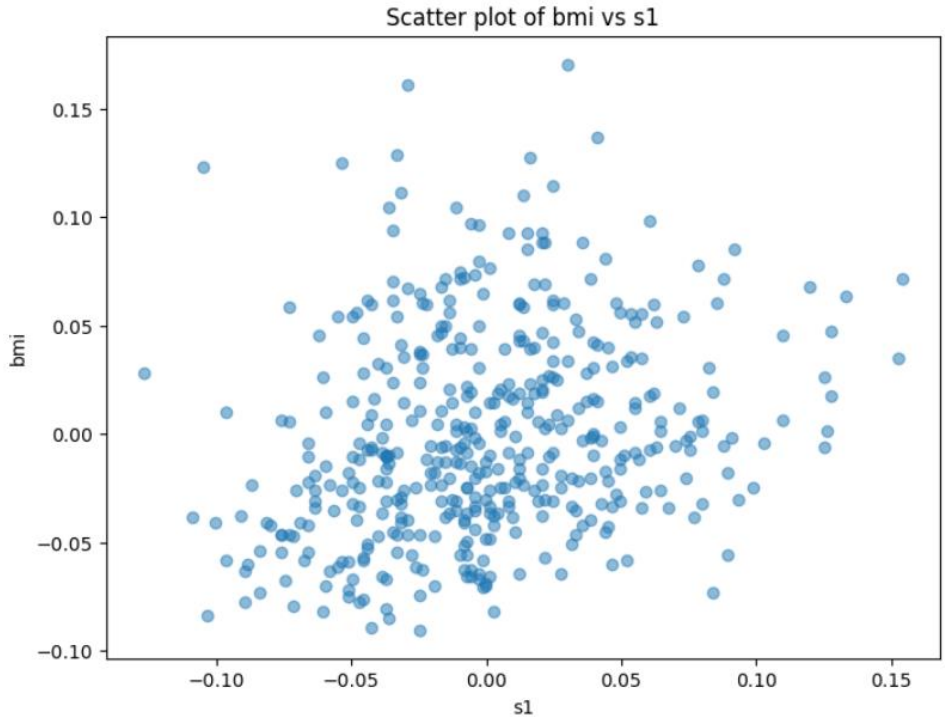
Для студентов группы ИУ5-25М, ИУ5И-25М, ИУ5И-26М - для произвольной колонки данных построить парные диаграммы (pairplot).

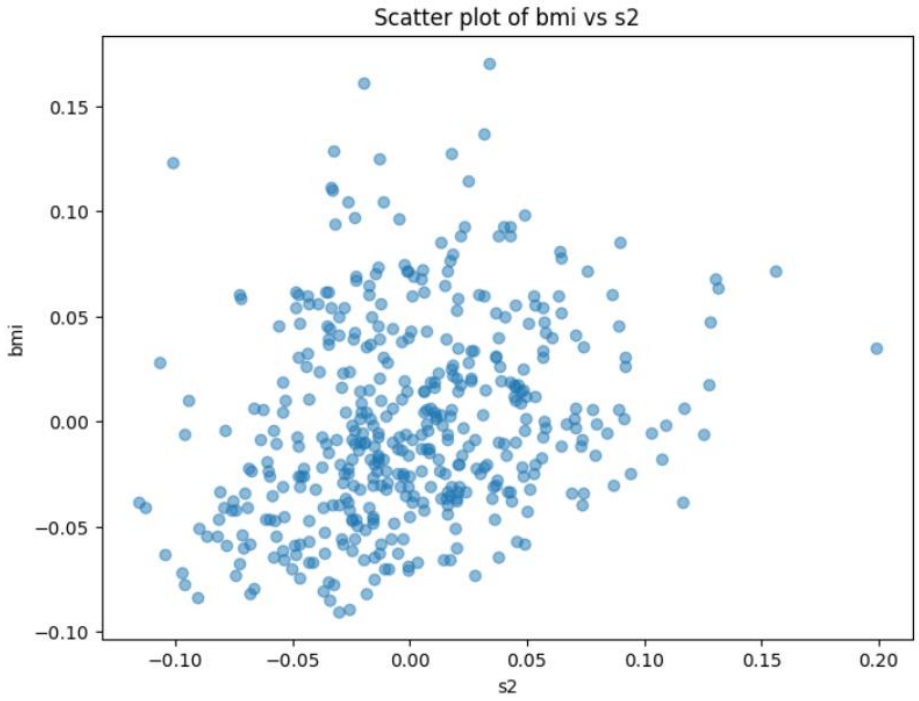
```python
#Дополнительные требования
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_diabetes
# Загрузка набора данных
diabetes_data = load_diabetes()
df = pd.DataFrame(data=diabetes_data.data, columns=diabetes_data.feature_names)
# Выбор произвольной колонки данных для построения парных диаграмм
column_name = 'bmi'
# Построение диаграммы рассеяния для выбранной колонки данных
for feature in df.columns:
    if feature != column_name:
        plt.figure(figsize=(8, 6))
        plt.scatter(df[feature], df[column_name], alpha=0.5)
        plt.xlabel(feature)
        plt.ylabel(column_name)
        plt.title(f"Scatter plot of {column_name} vs {feature}")
        plt.show()
```

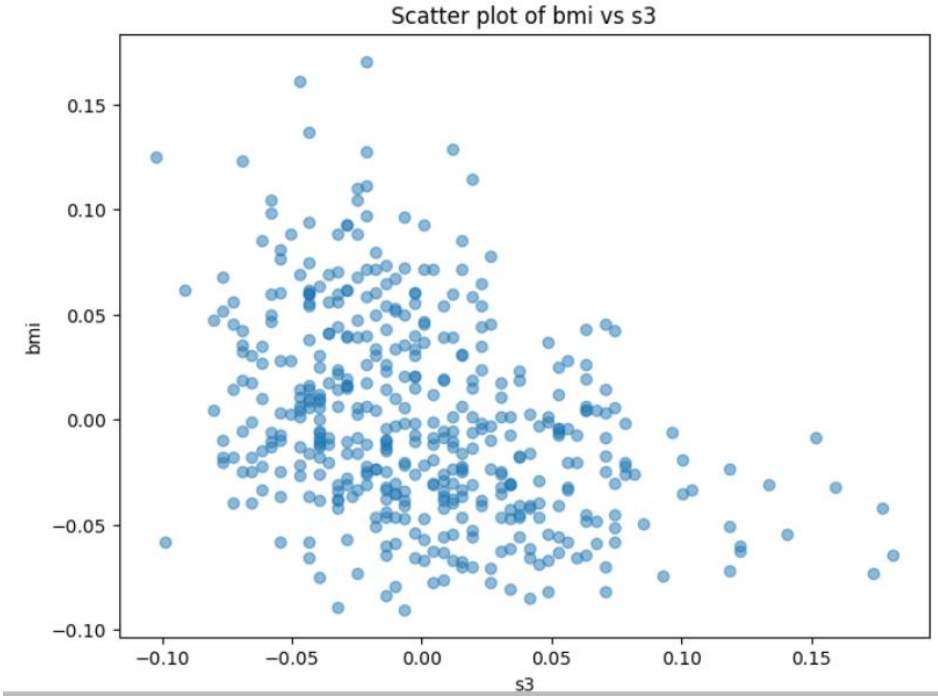


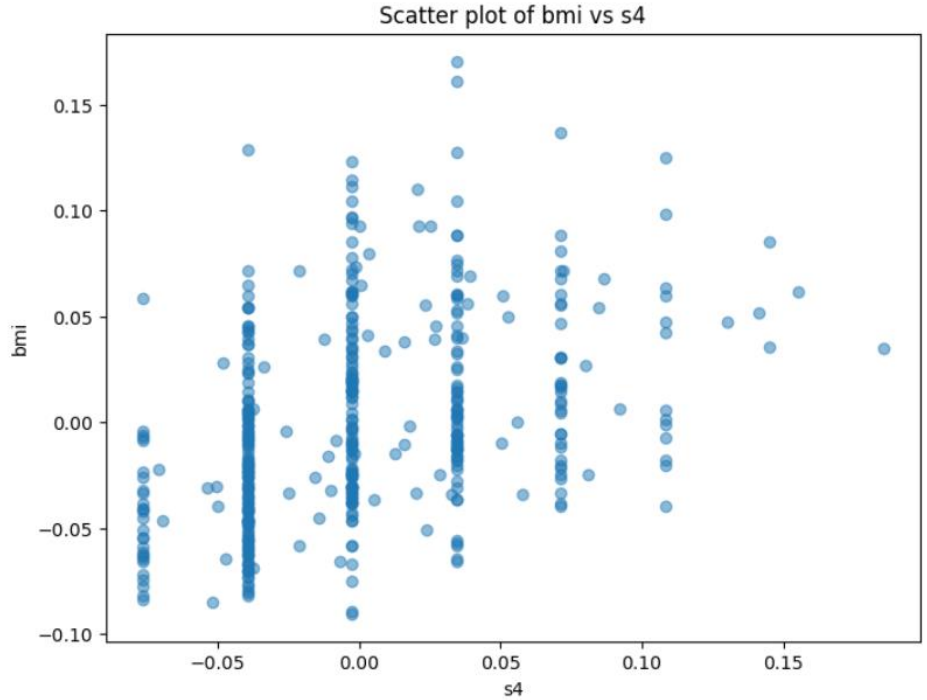


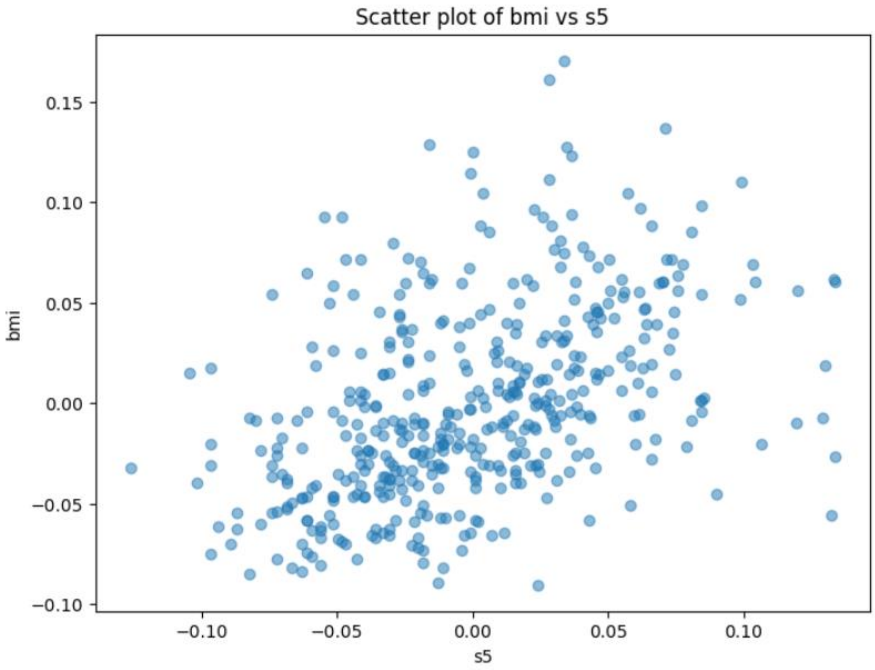

Scatter plot of bmi vs bp
bmi
bp
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.10
−0.05
0.00
0.05
0.10

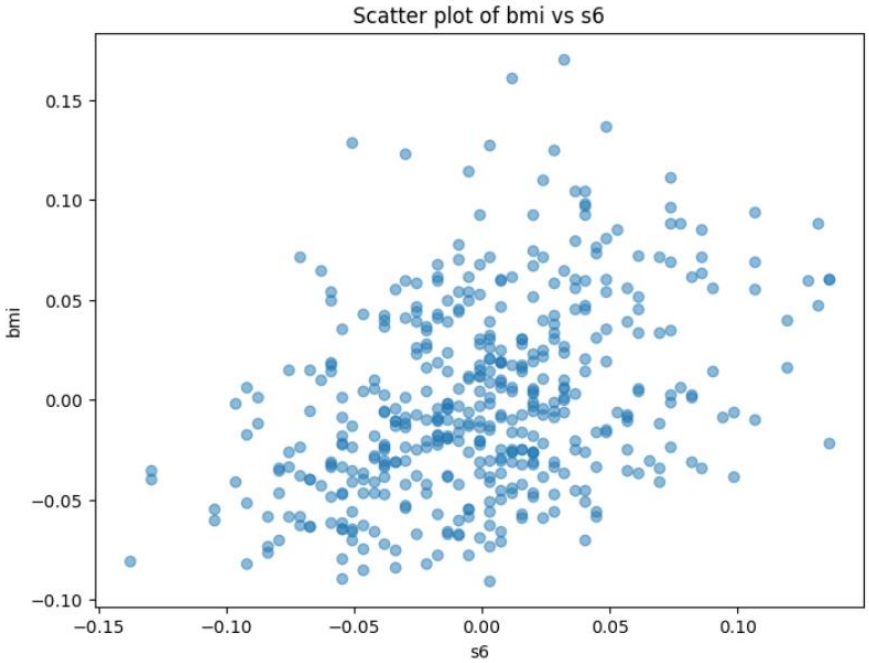

Scatter plot of bmi vs s1
bmi
s1
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.10
−0.05
0.00
0.05
0.10
0.15


Scatter plot of bmi vs s2
bmi
s2
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20


Scatter plot of bmi vs s3
bmi
s3
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.10
−0.05
0.00
0.05
0.10
0.15


Scatter plot of bmi vs s4
bmi
s4
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.05
0.00
0.05
0.10
0.15


Scatter plot of bmi vs s5
bmi
s5
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.10
−0.05
0.00
0.05
0.10


Scatter plot of bmi vs s6
bmi
s6
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.10
−0.05
0.00
0.05
0.10