Рубежный контроль №1 по курсу «Методы машинного обучения»

ИУ5-23М Бондаренко И. Г.

Вариант

- 3, 23 задание
- для произвольной колонки данных построить boxplot

Описание датасета

Stroke Prediction Dataset

Этот набор данных используется для прогнозирования вероятности инсульта у пациента на основе входных параметров, таких как пол, возраст, различные заболевания и статус курения. Каждая строка данных предоставляет соответствующую информацию о пациенте.

Информация об атрибутах 1) id: уникальный идентификатор

2) gender: "Мужской", "Женский" или "Другой"

3) age: возраст пациента

4) hypertension: 0, если у пациента нет гипертонии, 1, если у пациента есть гипертония

5) heart_disease: 0, если у пациента нет заболеваний сердца, 1, если у пациента есть заболевание сердца

6) ever_married: "Нет" или "Да"

7) work_type: "дети", "Государственный служащий", "Никогда не работал", "Частный" или "Самозанятый"

8) Residence_type: "Сельская местность" или "Город"

9) avg_glucose_level: средний уровень глюкозы в крови

10) bmi: индекс массы тела

11) smoking_status: "ранее курил", "никогда не курил", "курит" или "Неизвестно"*

12) stroke: 1, если у пациента был инсульт, или 0, если нет

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Подгрузим датасет и продемонстрируем его содержимое
data_loaded = pd.read_csv('dataset.csv', sep=",")
data_loaded.head()
```

ın	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	strol
0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	
0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	
0	1	Yes	Private	Rural	105.92	32.5	never smoked	
0	0	Yes	Private	Urban	171.23	34.4	smokes	
1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	

Next steps: View recommended plots

Задача 1. Для набора данных проведите кодирование одного (произвольного) категориального признака с использованием метода "weight of evidence (WoE) encoding".

```
# Функция для вычисления WoE для каждой категории
def calculate_woe(df, feature, target):
    total_good = df[target].sum()
    total_bad = len(df) - total_good
    category_woe = {}
    for category in df[feature].unique():
        good = df[(df[feature] == category) & (df[target] == 1)].shape[0]
        bad = df[(df[feature] == category) & (df[target] == 0)].shape[0]
        if good == 0:
            good = 0.5
        if bad == 0:
            bad = 0.5
        woe = (good / total_good) / (bad / total_bad)
        category_woe[category] = woe
    return category_woe

woe_encoding = calculate_woe(data_loaded, 'gender', 'stroke')
data_loaded['gender_WOE'] = data_loaded['gender'].map(woe_encoding)
data_loaded[['gender', 'gender_WOE']]
```

	gender	gender_WOE
0	Male	1.050516
1	Female	0.964814
2	Male	1.050516
3	Female	0.964814
4	Female	0.964814
...	...	...
5105	Female	0.964814
5106	Female	0.964814
5107	Female	0.964814
5108	Male	1.050516
5109	Female	0.964814

5110 rows × 2 columns

Задача 2. Для набора данных для одного (произвольного) числового признака проведите обнаружение и удаление выбросов на основе правила трех сигм.

```
def detect_outliers(data, threshold=3):
    mean = data.mean()
    std = data.std()
    lower_bound = mean - threshold * std
    upper_bound = mean + threshold * std
    return lower_bound, upper_bound

lower_bound, upper_bound = detect_outliers(data_loaded['avg_glucose_level'])

data_without_outliers = data_loaded[(data_loaded['avg_glucose_level'] >= lower_bound) & (data_loaded['avg
data_without_outliers
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type
0	9046	Male	67.0	0	1	Yes	Private
1	51676	Female	61.0	0	0	Yes	Self-employed
2	31112	Male	80.0	0	1	Yes	Private
3	60182	Female	49.0	0	0	Yes	Private
4	1665	Female	79.0	1	0	Yes	Self-employed
...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private
5106	44873	Female	81.0	0	0	Yes	Self-employed
5107	19723	Female	35.0	0	0	Yes	Self-employed
5108	37544	Male	51.0	0	0	Yes	Private
5109	44679	Female	44.0	0	0	Yes	Govt_job

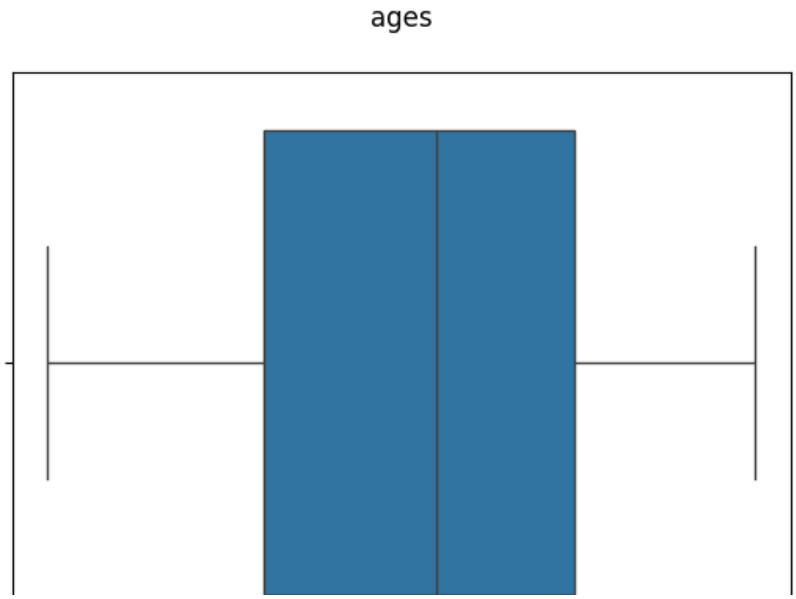
5061 rows × 13 columns

Next steps: View recommended plots

Boxplot

```
import seaborn as sns

sns.boxplot(data=data_loaded,x='age')
plt.title('ages\n')
plt.show()
```


ages