

Рубежный контроль №1 по курсу «Методы машинного обучения»

Подготовил: студент группы ИУ5-22М Ватулин И. А.

Вариант

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

Описание датасета

Mental Health Dataset - этот датасет содержит информацию из которой можно составить информацию о ментальном здоровье

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Подгрузим датасет и продемонстрируем его содержимое
data_loaded = pd.read_csv('mental.csv', sep=",")
data_loaded = data_loaded.drop('Timestamp', axis=1)
data_loaded.head()
```

	Gender	Country	Occupation	self_employed	family_history
treatment \					
0	Female	United States	Corporate	NaN	No
Yes					
1	Female	United States	Corporate	NaN	Yes
Yes					
2	Female	United States	Corporate	NaN	Yes
Yes					
3	Female	United States	Corporate	No	Yes
Yes					
4	Female	United States	Corporate	No	Yes
Yes					

	Days_Indoors	Growing_Stress	Changes_Habits	Mental_Health_History	\
0	1-14 days	Yes	No		Yes
1	1-14 days	Yes	No		Yes
2	1-14 days	Yes	No		Yes
3	1-14 days	Yes	No		Yes
4	1-14 days	Yes	No		Yes

	Mood_Swings	Coping_Struggles	Work_Interest	Social_Weakness	\
0	Medium	No	No	Yes	
1	Medium	No	No	Yes	

2	Medium	No	No	Yes
3	Medium	No	No	Yes
4	Medium	No	No	Yes

	mental_health_interview	care_options
0	No	Not sure
1	No	No
2	No	Yes
3	Maybe	Yes
4	No	Yes

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

```
# Преобразуем целевую переменную treatment из строкового формата в
# бинарный формат (0 - No, 1 - Yes)
data_loaded['treatment'] = data_loaded['treatment'].map({'No': 0,
'Yes': 1})

# 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 # Чтобы избежать деления на ноль
        if bad == 0:
            bad = 0.5 # Чтобы избежать деления на ноль
        woe = (good / total_good) / (bad / total_bad)
        category_woe[category] = woe
    return category_woe

# Вычисляем WoE для признака "Country"
woe_encoding_country = calculate_woe(data_loaded, 'Country',
'treatment')

# Применяем WoE кодирование к признаку "Country"
data_loaded['Country_WoE'] =
```

```
data_loaded['Country'].map(woe_encoding_country)
```

```
data_loaded[['Country', 'Country_WoE']]
```

	Country	Country_WoE
0	United States	1.165581
1	United States	1.165581
2	United States	1.165581
3	United States	1.165581
4	United States	1.165581
...
292359	United States	1.165581
292360	South Africa	1.526380
292361	United States	1.165581
292362	United States	1.165581
292363	United States	1.165581

```
[292364 rows x 2 columns]
```

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

```
# Подгрузим датасет и продемонстрируем его содержимое
```

```
data_loaded = pd.read_csv('spotify.csv', sep=";")
```

```
data_loaded = data_loaded.drop('id', axis=1)
```

```
data_loaded.head()
```

	artist_names	track_name \
0	ZAYN, PARTYNEXTDOOR	Still Got Time (feat. PARTYNEXTDOOR)
1	Alessia Cara	Growing Pains
2	The Killers	Mr. Brightside
3	Cardi B, Chance the Rapper	Best Life (feat. Chance The Rapper)
4	Post Malone, The Weeknd	One Right Now (with The Weeknd)

	source	key	mode	time_signature	danceability
energy \					
0	RCA Records Label	G	Major	4 beats	0.748
0.627					
1	Def Jam Recordings	C#/Db	Minor	4 beats	0.353
0.755					
2	Island Records	C#/Db	Major	4 beats	0.352
0.911					
3	Atlantic/KSR	A	Major	4 beats	0.620

```
0.625
4   Republic Records   C#/Db   Major           4 beats           0.687
0.781
```

	speechiness	acousticness	instrumentalness	liveness	valence	
loudness \						
0	0.0639	0.13100	0.0	0.0852	0.524	-
6.029						
1	0.7330	0.08220	0.0	0.3900	0.437	-
6.276						
2	0.0747	0.00121	0.0	0.0995	0.236	-
5.230						
3	0.5530	0.28700	0.0	0.3140	0.665	-
7.438						
4	0.0530	0.03610	0.0	0.0755	0.688	-
4.806						

	tempo	duration_ms	weeks_on_chart	streams
0	120.963	188491	17	107527761
1	191.153	193680	2	9944865
2	148.033	222973	125	512388123
3	167.911	284856	2	11985346
4	97.014	193507	30	301860377

```
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

# Вычислим границы для признака "duration_ms"
lower_bound, upper_bound = detect_outliers(data_loaded['duration_ms'])

# Удалим выбросы
data_without_outliers = data_loaded[(data_loaded['duration_ms'] >=
lower_bound) & (data_loaded['duration_ms'] <= upper_bound)]

data_without_outliers
```