Лабораторная работа №4

**Предобработка неискажённых данных**

# Цель:

Ознакомиться с методами предобработки данных из библиотеки *Scikit Learn*

# Выполнение:

## А. Загрузка данных

1. Загрузить датасет из прилагаемого файла (Данные представлены в виде csv таблицы). Можно скачать отсюда https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data.
2. Создать Python скрипт. Загрузить датасет в датафрейм, и исключить бинарные признаки и признак времени. Вывести датафрейм на консоль (простой print(df)).
3. Построить гистограммы признаков

import matplotlib.pyplot as plt

n\_bins = 20

fig, axs = plt.subplots(2,3)

axs[0, 0].hist(df['age'].values, bins = n\_bins) axs[0, 0].set\_title('age')

axs[0, 1].hist(df['creatinine\_phosphokinase'].values, bins = n\_bins) axs[0, 1].set\_title('creatinine\_phosphokinase')

axs[0, 2].hist(df['ejection\_fraction'].values, bins = n\_bins) axs[0, 2].set\_title('ejection\_fraction')

axs[1, 0].hist(df['platelets'].values, bins = n\_bins) axs[1, 0].set\_title('platelets')

axs[1, 1].hist(df['serum\_creatinine'].values, bins = n\_bins) axs[1, 1].set\_title('serum\_creatinine')

axs[1, 2].hist(df['serum\_sodium'].values, bins = n\_bins) axs[1, 2].set\_title('serum\_sodium')

plt.show()

1. На основании гистограмм определите диапазоны значений для каждого из признаков, а также возле какого значения лежит наибольшее количество наблюдений.
2. Так как библиотека Sklearn работает с NumPy массива, то преобразуйте датафрейм к двумерному массиву NumPy, где строка соответствует наблюдению, а столбец признаку

data = df.to\_numpy(dtype='float')

## В. Стандартизация данных

1. Подключите модуль Sklearn. Настройте стандартизацию на основе первых 150 наблюдений используя StandardScaler

from sklearn import preprocessing

scaler = preprocessing.StandardScaler().fit(data[:150,:])

1. Стандартизуйте все данные

data\_scaled = scaler.transform(data)

1. Постройте гистограммы стандартизированных данных

fig, axs = plt.subplots(2,3)

axs[0, 0].hist(data\_scaled[:,0], bins = n\_bins) axs[0, 0].set\_title('age')

axs[0, 1].hist(data\_scaled[:,1], bins = n\_bins) axs[0, 1].set\_title('creatinine\_phosphokinase')

axs[0, 2].hist(data\_scaled[:,2], bins = n\_bins) axs[0, 2].set\_title('ejection\_fraction')

axs[1, 0].hist(data\_scaled[:,3], bins = n\_bins) axs[1, 0].set\_title('platelets')

axs[1, 1].hist(data\_scaled[:,4], bins = n\_bins) axs[1, 1].set\_title('serum\_creatinine')

axs[1, 2].hist(data\_scaled[:,5], bins = n\_bins) axs[1, 2].set\_title('serum\_sodium')

plt.show()

1. Сравните данные до и после стандартизации. Опишите, что изменилось и почему.
2. Рассчитайте мат. ожидание и СКО до и после стандартизации. На основании этих значений выведите для каждого признака формулы по которым они стандартизировались.
3. Сравните значений из формул с полями *mean\_* и *var\_* объекта *scaler*
4. Проведите настройку стандартизации на всех данных и сравните с результатами настройки на основании 150 наблюдений

**Примечание: вместо двух методов fit и transform можно использовать метод**

**fit\_transfrom, чтобы сразу настроить параметры и преобразовать данные.**

## С. Приведение к диапазону

1. Приведите данные к диапазону [0,1], используя MinMaxScaler

min\_max\_scaler = preprocessing.MinMaxScaler().fit(data) data\_min\_max\_scaled = min\_max\_scaler.transform(data)

1. Постройте гистограммы для признаков и сравните с исходными данными
2. Через параметры *MinMaxScaler* определите минимальное и максимальное значение в данных для каждого признака
3. Напишите функцию, которая приводит все данные к диапазону [-5 10]. Примените её и постройте гистограммы полученных данных.

## D. Нелинейные преобразования

1. Приведите данные к равномерному распределению используя **QuantileTransformer**

quantile\_transformer = preprocessing.QuantileTransformer(n\_quantiles = 100, random\_state=0).fit(data)

data\_quantile\_scaled = quantile\_transformer.transform(data)

1. Постройте гистограммы и сравните с исходными данными
2. Определите, как и на что влияет значение параметра *n\_quantiles*
3. Приведите данные к нормальному распределению передав в *QuantileTransformer* параметр

*output\_distribution=‘normal’*

1. Постройте гистограммы и сравните с исходными данными
2. Самостоятельно приведите данные к нормальному распределению используя PowerTransformer. Приведите скрипт операции и гистограммы полученных данных.

## E. Дискретизация признаков

1. Проведите дискретизацию признаков, используя KBinsDiscretizer, на следующее количество диапазонов:

age - 5

creatinine\_phosphokinase - 4

ejection\_fraction - 6

platelets - 10

serum\_creatinine - 7

serum\_sodium - 4

1. Постройте гистограммы. Объясните полученные результаты
2. Через параметр *bin\_edges\_* выведите диапазоны каждого интервала для каждого признака