Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления (ИУ5)

О т ч е т

по лабораторной работе №1

«Разведочный анализ данных.

Исследование и визуализация данных.»

Дисциплина: Технологии машинного обучения

Студент гр. ИУ5-63Б _____ Назаров М.М.

(Подпись, дата) (И.О. Фамилия)

Преподаватель _____ Гапанюк Ю.Е.

(Подпись, дата) (И.О. Фамилия)

Москва, 2021

Разведочный анализ данных. Исследование и визуализация данных.

1)Текстовое описание набора данных

В качестве набора данных я использован данные о прогноз сердечной недостаточности - https://www.kaggle.com/andrewmvd/heart-failure-clinical-data

Описание набора данных с сайта: Сердечно-сосудистые заболевания (ССЗ) являются причиной смерти номер 1 во всем мире, ежегодно унося около 17,9 миллиона жизней, что составляет 31% всех смертей в мире. Сердечная недостаточность - частое явление, вызываемое сердечно-сосудистыми заболеваниями, и этот набор данных содержит 12 функций, которые можно использовать для прогнозирования смертности от сердечной недостаточности. Большинство сердечно-сосудистых заболеваний можно предотвратить путем устранения поведенческих факторов риска, таких как употребление табака, нездоровое питание и ожирение, недостаточная физическая активность и вредное употребление алкоголя, с использованием стратегий, охватывающих все население. Людям с сердечно-сосудистыми заболеваниями или с высоким риском сердечно-сосудистых заболеваний (из-за наличия одного или нескольких факторов риска, таких как гипертония, диабет, гиперлипидемия или уже установленное заболевание) необходимо раннее выявление и лечение, при этом модель машинного обучения может оказаться очень полезной.

Импорт библиотек

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

Загрузка данных

In [3]:
```
data = pd.read_csv('heart_failure_clinical_records_dataset.csv', sep=",")
```

2)Основные характеристики датасета.

Первые 5 строчек датасета

In [6]:
```
data.head()
```

Out[6]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platele
0	75.0	0	582	0	20	1	265000.(
1	55.0	0	7861	0	38	0	263358.(
2	65.0	0	146	0	20	0	162000.(

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platele
3	50.0	1	111	0	20	0	210000.(

Размер датасета - 299 строк и 13 столбцов

In [7]:
```
data.shape
```

Out[7]:
```
(299, 13)
```

Список всех столбоц датасета

In [8]:
```
data.columns
```

Out[8]:
```
Index(['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes',
       'ejection_fraction', 'high_blood_pressure', 'platelets',
       'serum_creatinine', 'serum_sodium', 'sex', 'smoking', 'time',
       'DEATH_EVENT'],
      dtype='object')
```

Список всех столбцов с их типами данных

In [9]:
```
data.dtypes
```

Out[9]:
```
age                         float64
anaemia                       int64
creatinine_phosphokinase      int64
diabetes                      int64
ejection_fraction             int64
high_blood_pressure           int64
platelets                   float64
serum_creatinine            float64
serum_sodium                  int64
sex                           int64
smoking                       int64
time                          int64
DEATH_EVENT                   int64
dtype: object
```

Проверка на наличие пустых ячеек в датасете - в данном датасете нет пустых значений

In [10]:
```
for col in data.columns:
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
age - 0
anaemia - 0
creatinine_phosphokinase - 0
diabetes - 0
ejection_fraction - 0
high_blood_pressure - 0
platelets - 0
serum_creatinine - 0
serum_sodium - 0
sex - 0
smoking - 0
time - 0
DEATH_EVENT - 0
```

Основные статистические характеристики датасета

In [11]:
```
data.describe()
```

Out[11]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pr
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.0
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.3
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.4
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.0
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.0
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.0
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.0
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.0

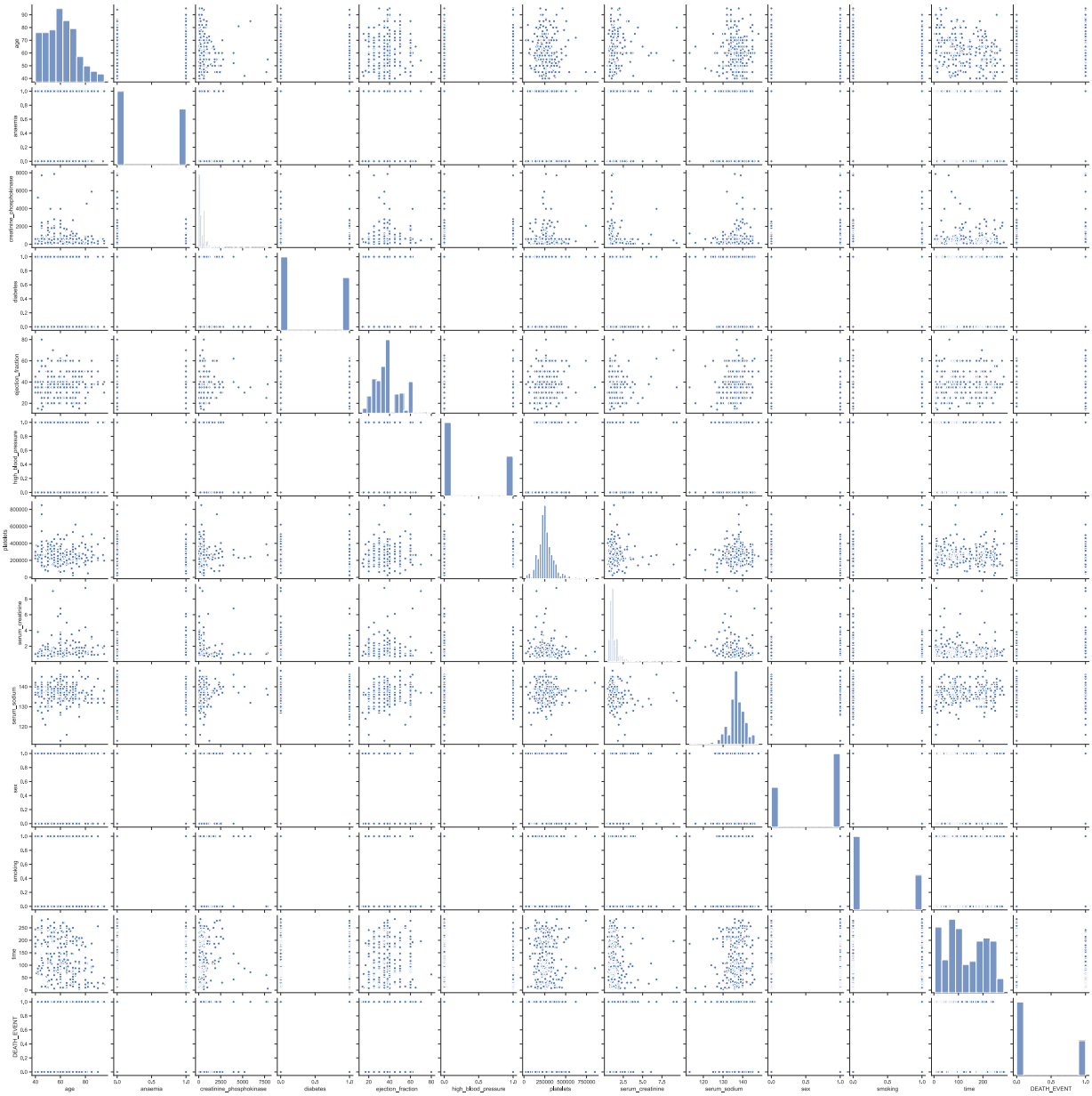
3)Визуальное исследование датасета.

In [12]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='age', y='serum_sodium', data=data)
```

Out[12]:
```
<AxesSubplot:xlabel='age', ylabel='serum_sodium'>
```

In [13]:
```
sns.pairplot(data)
```

Out[13]:
```
<seaborn.axisgrid.PairGrid at 0x18357c5dbe0>
```



In [14]:
```
sns.pairplot(data, hue="DEATH_EVENT")
```

Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x1835e234d30>
```