|  |  |
| --- | --- |
|  | **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

КАФЕДРА                  Системы обработки информации и управления

**Рубежный контроль №1**

**По курсу «Технологии машинного обучения»**

**Вариант 11**

Подготовила:

Студентка группы ИУ5-65Б.

Очеретная С.В.

08.04.2022

Проверил:

Преподаватель кафедры ИУ5  
Гапанюк Ю.Е.

Москва, 2022 г.

# **Тема**: Технологии разведочного анализа и обработки данных.



**Дополнительные требования по группам:**

* Для студентов группы ИУ5-65Б - для набора данных построить "парные диаграммы".

## **Задача №2.**

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

### Загрузка библиотек и датасета

Набор данных: [датасет 3](https://www.kaggle.com/datasets/fivethirtyeight/fivethirtyeight-comic-characters-dataset)

import numpy as np  
import pandas as pd  
import seaborn as sns  
from sklearn.impute import SimpleImputer  
import matplotlib.pyplot as plt  
%matplotlib inline   
sns.set(style="ticks")

filename = '../datasets/marvel-wikia-data.csv'  
data = pd.read\_csv(filename)  
data.head()

page\_id name \  
0 1678 Spider-Man (Peter Parker)   
1 7139 Captain America (Steven Rogers)   
2 64786 Wolverine (James \"Logan\" Howlett)   
3 1868 Iron Man (Anthony \"Tony\" Stark)   
4 2460 Thor (Thor Odinson)   
  
 urlslug ID \  
0 \/Spider-Man\_(Peter\_Parker) Secret Identity   
1 \/Captain\_America\_(Steven\_Rogers) Public Identity   
2 \/Wolverine\_(James\_%22Logan%22\_Howlett) Public Identity   
3 \/Iron\_Man\_(Anthony\_%22Tony%22\_Stark) Public Identity   
4 \/Thor\_(Thor\_Odinson) No Dual Identity   
  
 ALIGN EYE HAIR SEX GSM \  
0 Good Characters Hazel Eyes Brown Hair Male Characters NaN   
1 Good Characters Blue Eyes White Hair Male Characters NaN   
2 Neutral Characters Blue Eyes Black Hair Male Characters NaN   
3 Good Characters Blue Eyes Black Hair Male Characters NaN   
4 Good Characters Blue Eyes Blond Hair Male Characters NaN   
  
 ALIVE APPEARANCES FIRST APPEARANCE Year   
0 Living Characters 4043.0 Aug-62 1962.0   
1 Living Characters 3360.0 Mar-41 1941.0   
2 Living Characters 3061.0 Oct-74 1974.0   
3 Living Characters 2961.0 Mar-63 1963.0   
4 Living Characters 2258.0 Nov-50 1950.0

### Обработка пропусков

data.shape

(16376, 13)

# пропущенные значения  
data.isnull().sum()

page\_id 0  
name 0  
urlslug 0  
ID 3770  
ALIGN 2812  
EYE 9767  
HAIR 4264  
SEX 854  
GSM 16286  
ALIVE 3  
APPEARANCES 1096  
FIRST APPEARANCE 815  
Year 815  
dtype: int64

data.dtypes

page\_id int64  
name object  
urlslug object  
ID object  
ALIGN object  
EYE object  
HAIR object  
SEX object  
GSM object  
ALIVE object  
APPEARANCES float64  
FIRST APPEARANCE object  
Year float64  
dtype: object

data\_clean = data

**Категориальные признаки:**

# Импьютация константой NA колонки ID  
imp\_id = SimpleImputer(missing\_values=np.nan, strategy='constant', fill\_value='NA')  
data\_clean[['ID']] = imp\_id.fit\_transform(data\_clean[['ID']])

# Импьютация константой NA колонки ALIGN  
imp\_ALIGN = SimpleImputer(missing\_values=np.nan, strategy='constant', fill\_value='NA')  
data\_clean[['ALIGN']] = imp\_ALIGN.fit\_transform(data\_clean[['ALIGN']])

# Импьютация самым частым колонки EYE  
imp\_EYE = SimpleImputer(missing\_values=np.nan, strategy='most\_frequent')  
data\_clean[['EYE']] = imp\_EYE.fit\_transform(data\_clean[['EYE']])

# Импьютация самым частым колонки HAIR  
imp\_HAIR = SimpleImputer(missing\_values=np.nan, strategy='most\_frequent')  
data\_clean[['HAIR']] = imp\_HAIR.fit\_transform(data\_clean[['HAIR']])

# Импьютация константой NA колонки SEX  
imp\_SEX = SimpleImputer(missing\_values=np.nan, strategy='constant', fill\_value='NA')  
data\_clean[['SEX']] = imp\_SEX.fit\_transform(data\_clean[['SEX']])

# Удалим колонку GSM, т.к. она почти пустая  
data\_clean = data\_clean.drop(columns = ['GSM'], axis = 1)

# Импьютация константой NA колонки ALIVE  
imp\_ALIVE = SimpleImputer(missing\_values=np.nan, strategy='constant', fill\_value='NA')  
data\_clean[['ALIVE']] = imp\_ALIVE.fit\_transform(data\_clean[['ALIVE']])

# Импьютация константой NA колонки FIRST APPEARANCE  
imp\_FIRST\_APPEARANCE = SimpleImputer(missing\_values=np.nan, strategy='constant', fill\_value='NA')  
data\_clean[['FIRST APPEARANCE']] = imp\_FIRST\_APPEARANCE.fit\_transform(data\_clean[['FIRST APPEARANCE']])

**Количественные признаки:**

strategies=['median', 'most\_frequent']

# импьютация нужной колонки с помощью нужной стратегии  
def func\_impute\_col(dataset, column, strategy\_param):  
 temp\_data = dataset[[column]]  
   
 imp\_num = SimpleImputer(strategy=strategy\_param)  
 data\_num\_imp = imp\_num.fit\_transform(temp\_data)  
   
 return data\_num\_imp

# замена медианой APPEARANCES  
col\_APPEARANCES\_imp = func\_impute\_col(data\_clean, 'APPEARANCES', strategies[0])  
data\_clean[['APPEARANCES']] = col\_APPEARANCES\_imp

# замена часто встречаемым Year  
col\_Year\_imp = func\_impute\_col(data\_clean, 'Year', strategies[1])  
data\_clean[['Year']] = col\_Year\_imp

data\_clean.isnull().sum()

page\_id 0  
name 0  
urlslug 0  
ID 0  
ALIGN 0  
EYE 0  
HAIR 0  
SEX 0  
ALIVE 0  
APPEARANCES 0  
FIRST APPEARANCE 0  
Year 0  
dtype: int64

### Ответы на вопросы

Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали?

* замена константой (constant) - для категориальных
* замена самым часто встречаемым значением (most\_frequent) - для категориальных и количественных
* замена медианой - для количественных

Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

* колонка GSM использоваться не будет, т.к. содержит очень много пропущенных значений
* столбец EYE тоже бы удалила из-за большого количества пропусков
* остальные признаки необходимы для описания датасета, поэтому их бы я отставила
* большинство признаков являются строковыми, а не числовыми, поэтому по корреляционной матрице отсеивать не будем