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»

Отчет по лабораторной работе №2

«Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных»

по дисциплине «Технологии машинного обучения»

Выполнил:
студент группы ИУ5Ц-84Б
Тихонова Д.Д.
подпись, дата

Проверил:
к.т.н., доц., Ю.Е. Гапанюк
подпись, дата

2025 г.

1. Текст программы

ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

анды + Код + Текст

Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных.

```
from google.colab import files
files.upload()

!pip install -q kaggle

!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json

!kaggle datasets download -d ronaldonyango/global-suicide-rates-1990-to-2022

!unzip global-suicide-rates-1990-to-2022.zip

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
data = pd.read_csv('suicide_rates_1990-2022.csv', sep=",")
print(data.head())
```

Выбрать файлы kaggle.json

- **kaggle.json**(application/json) - 67 bytes, last modified: 17.04.2025 - 100% done

```
Saving kaggle.json to kaggle.json
```

ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

анды + Код + Текст

90 % Сбросить

```
License(s): other
Archive:  global-suicide-rates-1990-to-2022.zip
  inflating: age_std_suicide_rates_1990-2022.csv
  inflating: suicide_rates_1990-2022.csv
  RegionCode RegionName CountryCode CountryName  Year   Sex     AgeGroup  \
0         EU     Europe         ALB     Albania  1992  Male   0-14 years
1         EU     Europe         ALB     Albania  1992  Male   0-14 years
2         EU     Europe         ALB     Albania  1992  Male   0-14 years
3         EU     Europe         ALB     Albania  1992  Male   0-14 years
4         EU     Europe         ALB     Albania  1992  Male  15-24 years

        Generation  SuicideCount  CauseSpecificDeathPercentage  \
0  Generation Alpha           0.0                      0.000000
1  Generation Alpha           0.0                      0.000000
2  Generation Alpha           0.0                      0.000000
3  Generation Alpha           0.0                      0.000000
4     Generation Z            5.0                      3.401361

   DeathRatePer100K  Population          GDP  GDPPerCapita  \
0          0.000000   3247039.0  652174990.8     200.85222
1          0.000000   3247039.0  652174990.8     200.85222
2          0.000000   3247039.0  652174990.8     200.85222
3          0.000000   3247039.0  652174990.8     200.85222
4          3.531073   3247039.0  652174990.8     200.85222

   GrossNationalIncome  GNIPerCapita  InflationRate  EmploymentPopulationRatio
0          906184212.3        1740.0     226.005421                     45.315
1          906184212.3        1740.0     226.005421                     45.315
2          906184212.3        1740.0     226.005421                     45.315
3          906184212.3        1740.0     226.005421                     45.315
4          906184212.3        1740.0     226.005421                     45.315
```

```
[2]  data.shape
```

```
(118560, 18)
```

```
[3]  data.dtypes
```

ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

+ Код + Текст

```
data.dtypes
```

	0
RegionCode	object
RegionName	object
CountryCode	object
CountryName	object
Year	int64
Sex	object
AgeGroup	object
Generation	object
SuicideCount	float64
CauseSpecificDeathPercentage	float64
DeathRatePer100K	float64
Population	float64
GDP	float64
GDPPerCapita	float64
GrossNationalIncome	float64
GNIPerCapita	float64
InflationRate	float64
EmploymentPopulationRatio	float64

dtype: object

ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

+ Код + Текст

```
# проверим есть ли пропущенные значения
data.isnull().sum()
```

	0
RegionCode	0
RegionName	0
CountryCode	0
CountryName	0
Year	0
Sex	0
AgeGroup	0
Generation	0
SuicideCount	464
CauseSpecificDeathPercentage	4289
DeathRatePer100K	10664
Population	5920
GDP	7240
GDPPerCapita	7240
GrossNationalIncome	9960
GNIPerCapita	10760
InflationRate	14460
EmploymentPopulationRatio	11120

dtype: int64

+ Код + Текст

dtype: int64

```
# Первые 5 строк датасета
data.head()
```

	RegionCode	RegionName	CountryCode	CountryName	Year	Sex	AgeGroup	Generation	SuicideCount	CauseSpecificDeathPercentage	DeathRatePer100K	Popu
0	EU	Europe	ALB	Albania	1992	Male	0-14 years	Generation Alpha	0.0	0.000000	0.000000	324
1	EU	Europe	ALB	Albania	1992	Male	0-14 years	Generation Alpha	0.0	0.000000	0.000000	324
2	EU	Europe	ALB	Albania	1992	Male	0-14 years	Generation Alpha	0.0	0.000000	0.000000	324
3	EU	Europe	ALB	Albania	1992	Male	0-14 years	Generation Alpha	0.0	0.000000	0.000000	324
4	EU	Europe	ALB	Albania	1992	Male	15-24 years	Generation Z	5.0	3.401361	3.531073	324

```
[9] total_count = data.shape[0]
    print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 118560
```

Обработка пропусков в данных

Простые стратегии - удаление или заполнение нулями

```
[10] # Удаление колонок, содержащих пустые значения
```

ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb
Файл Изменить Вид Вставка Среда выполнения Инструменты Справка
+ Код + Текст

"Внедрение значений" - импьютация (imputation)

Обработка пропусков в числовых данных

```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col, dt, temp_null_count, temp_perc))
```

```
Колонка SuicideCount. Тип данных float64. Количество пустых значений 464, 0.39%.
Колонка CauseSpecificDeathPercentage. Тип данных float64. Количество пустых значений 4289, 3.62%.
Колонка DeathRatePer100K. Тип данных float64. Количество пустых значений 10664, 8.99%.
Колонка Population. Тип данных float64. Количество пустых значений 5920, 4.99%.
Колонка GDP. Тип данных float64. Количество пустых значений 7240, 6.11%.
Колонка GDPPerCapita. Тип данных float64. Количество пустых значений 7240, 6.11%.
Колонка GrossNationalIncome. Тип данных float64. Количество пустых значений 9960, 8.4%.
Колонка GNIPerCapita. Тип данных float64. Количество пустых значений 10760, 9.08%.
Колонка InflationRate. Тип данных float64. Количество пустых значений 14460, 12.2%.
Колонка EmploymentPopulationRatio. Тип данных float64. Количество пустых значений 11120, 9.38%.
```

```
[14] # Фильтр по колонкам с пропущенными значениями
     data_num = data[num_cols]
     data_num
```

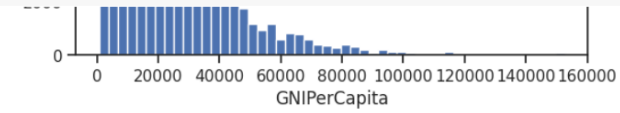
	SuicideCount	CauseSpecificDeathPercentage	DeathRatePer100K	Population	GDP	GDPPerCapita	GrossNationalIncome	GNIPerCapita	Inflati
0	0.0	0.000000	0.000000	3247039.0	6.521750e+08	200.85222	0.061842e+08	1740.0	226

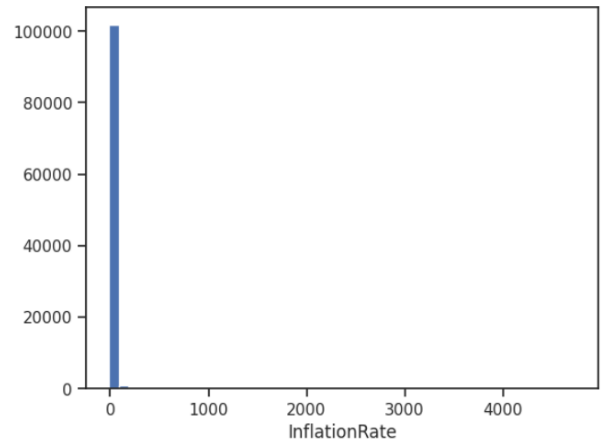
ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

+ Код + Текст

```
for col in data_num:
    plt.hist(data[col], 50)
    plt.xlabel(col)
    plt.show()
```





ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

+ Код + Текст

[16]
```
data_num_MasVnrArea = data_num[['SuicideCount']]
data_num_MasVnrArea.head()
```

	SuicideCount
0	0.0
1	0.0
2	0.0
3	0.0
4	5.0

[17]
```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

```
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_num_MasVnrArea)
mask_missing_values_only
```

```
array([[False],
       [False],
       [False],
       ...,
       [False],
       [False],
       [False]])
```

+ Код + Текст

[ ]
```
strategies=['mean', 'median', 'most_frequent']
```

[ ]
```
def test_num_impute(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
```

ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

ынды + Код + Текст

strategies[1], test_num_impute(strategies[1])

```python
strategies[2], test_num_impute(strategies[2])
```

```
('most_frequent',
 array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0.]))
```

```python
[ ] # Более сложная функция, которая позволяет задавать колонку и вид импьютации
def test_num_impute_col(dataset, column, strategy_param):
    temp_data = dataset[[column]]
```

ИУ5Ц_84Б_Тихонова_Лаб_2.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

ынды + Код + Текст

Обработка пропусков в категориальных данных

```python
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
cat_cols = []
for col in data.columns:
    dt = str(data[col].dtype)
    if (dt!='object'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}.'.format(col, dt))
```

```
Колонка Year. Тип данных int64.
Колонка SuicideCount. Тип данных float64.
Колонка CauseSpecificDeathPercentage. Тип данных float64.
Колонка DeathRatePer100K. Тип данных float64.
Колонка Population. Тип данных float64.
Колонка GDP. Тип данных float64.
Колонка GDPPerCapita. Тип данных float64.
Колонка GrossNationalIncome. Тип данных float64.
Колонка GNIPerCapita. Тип данных float64.
Колонка InflationRate. Тип данных float64.
Колонка EmploymentPopulationRatio. Тип данных float64.
```

```python
[ ] cat_temp_data = data[['RegionName']]
cat_temp_data.head()
```

	RegionName
0	Europe
1	Europe
2	Europe
3	Europe
4	Europe