

**Московский государственный технический
университет им. Н. Э. Баумана**

Курс «Технологии машинного обучения»

Отчёт по лабораторной работе №2

Выполнил:
Флоринский В. А.
группа ИУ5-64Б

Проверил:
Гапанюк Ю.Е.

Дата: 07.04.25

Дата:

Подпись:

Подпись:

Москва, 2025 г.

Цель лабораторной работы: изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Задание:

1. Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)
2. Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:
 - a. обработку пропусков в данных;
 - b. кодирование категориальных признаков;
 - c. масштабирование данных.

Ход выполнения:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
%matplotlib inline
```

Python

```
data = pd.read_csv('cleaned_blood_type_distribution_by_country.csv', sep=",")
```

Python

```
data.head()
```

Python

	Country/Dependency	Population	O+	A+	B+	AB+	O-	A-	B-	AB-
0	Albania	3,074,579	34.10%	31.20%	14.50%	5.20%	6.00%	5.50%	2.60%	0.90%
1	Algeria	43,576,691	40.00%	30.00%	15.00%	4.25%	6.60%	2.30%	1.10%	0.75%
2	Argentina	45,479,118	50.34%	31.09%	8.20%	2.16%	4.29%	2.98%	0.74%	0.20%
3	Armenia	3,021,324	29.00%	46.30%	12.00%	5.60%	2.00%	3.70%	1.00%	0.40%
4	Australia	25,466,459	38.00%	32.00%	12.00%	4.00%	7.00%	6.00%	2.00%	1.00%

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

Python

Всего строк: 126

```
> data.columns
```

Python

[5]

```
.. Index(['Country/Dependency', 'Population', 'O+', 'A+', 'B+', 'AB+', 'O-', 'A-',  
        'B-', 'AB-'],  
        dtype='object')
```

```
> print(data.isnull().sum())
```

Python

[6]

```
.. Country/Dependency    0  
   Population            0  
   O+                   0  
   A+                   0  
   B+                   0  
   AB+                  1  
   O-                   1  
   A-                   1  
   B-                   1  
   AB-                  1  
   dtype: int64
```

1.

```
data_cleaned = data.dropna()
```

Python

[7]

```
print(data_cleaned.isnull().sum())
```

Python

```
Country/Dependency    0
Population             0
0+                    0
A+                    0
B+                    0
AB+                   0
0-                    0
A-                    0
B-                    0
AB-                   0
dtype: int64
```

2. модой возраст

```
data['AB+'] = data['AB+'].fillna(data['A+'].mode()[0]) # Заполнение модой
```

Python

+ Code

+ Markdown

3) категориальные другие значения

```
data['Country/Dependency'] = data['Country/Dependency'].fillna('Неизвестно')
```

Python

```
# Использование get_dummies для преобразования категориальных признаков
data = pd.get_dummies(data, columns=['Country/Dependency'], drop_first=True)

print(data.head())
```

Python

	Population	0+	A+	B+	AB+	0-	A-	B-	AB-	\
0	3,074,579	34.10%	31.20%	14.50%	5.20%	6.00%	5.50%	2.60%	0.90%	
1	43,576,691	40.00%	30.00%	15.00%	4.25%	6.60%	2.30%	1.10%	0.75%	
2	45,479,118	50.34%	31.09%	8.20%	2.16%	4.29%	2.98%	0.74%	0.20%	
3	3,021,324	29.00%	46.30%	12.00%	5.60%	2.00%	3.70%	1.00%	0.40%	
4	25,466,459	38.00%	32.00%	12.00%	4.00%	7.00%	6.00%	2.00%	1.00%	

	Country/Dependency_Algeria	...	Country/Dependency_Ukraine	\
0	False	...	False	
1	True	...	False	
2	False	...	False	
3	False	...	False	
4	False	...	False	

	Country/Dependency_United Arab Emirates	Country/Dependency_United Kingdom	\
0	False	False	
1	False	False	
2	False	False	
3	False	False	
4	False	False	

	Country/Dependency_United States	Country/Dependency_Uzbekistan	\
0	False	False	
1	False	False	
2	False	False	
...			
3	False		
4	False		

```
data_cleaned['0+'] = data['0+'].str.rstrip('%').astype(float) / 100
```

Python

```
data_cleaned['Population'] = data['Population'].str.replace(',', '').astype(float)
```

[20]

Python

```
scaler = MinMaxScaler()
```

```
data_cleaned[['0+', 'Population']] = scaler.fit_transform(data_cleaned[['0+', 'Population']])
```

[21]

Python

```
print(data_cleaned.head())
```

[22]

Python

```
... Country/Dependency Population      0+      A+      B+      AB+      0-  \
0      Albania      0.000391  0.147917  31.20%  14.50%  5.20%  6.00%
1      Algeria      0.005601  0.270833  30.00%  15.00%  4.25%  6.60%
2      Argentina      0.005846  0.486250  31.09%   8.20%  2.16%  4.29%
3      Armenia      0.000384  0.041667  46.30%  12.00%  5.60%  2.00%
4      Australia      0.003271  0.229167  32.00%  12.00%  4.00%  7.00%

      A-      B-      AB-
0  5.50%  2.60%  0.90%
1  2.30%  1.10%  0.75%
2  2.98%  0.74%  0.20%
3  3.70%  1.00%  0.40%
4  6.00%  2.00%  1.00%
```

```
plt.hist(data['Population'], 50)
plt.show()
```

[23]

Python

