



**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 вычислительная техника»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №2
«Обработка пропусков в данных, кодирование категориальных признаков,
масштабирование данных»

Выполнил:
студент группы ИУ5-63Б

Баркалова И.В.

Подпись и дата:

Проверил:
преподаватель каф.
ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2022 г.

Цель лабораторной работы

Изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Описание задания

1. Выбрать набор данных, содержащий категориальные признаки и пропуски данных.
2. Для выбранного датасета на основе материалов лекции решить следующие задачи:
 - 1) обработку пропусков в данных
 - 2) кодирование категориальных признаков
 - 3) масштабирование данных

Текст программы и результаты ее выполнения

```
In [3]: #Импортируем необходимые библиотеки  
import numpy as np  
import pandas as pd  
from sklearn.datasets import *  
import seaborn as sns  
import matplotlib.pyplot as plt  
%matplotlib inline  
sns.set(style="ticks")
```

```
In [4]: wine = load_wine()
```

```
In [5]: type(wine)
```

Out[5]:

```
In [8]: data = pd.DataFrame(data=np.c_[wine['data'], wine['target']],  
                           columns= wine['feature_names']+[ 'target'])
```

```
In [9]: data
```

```
Out[9]:
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavai
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	
...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	

178 rows × 14 columns



```
In [10]: data.shape
```

Out[10]:

```
In [11]: data.dtypes
```

Out[11]:

```
In [12]: data.isna()
```

Out[12]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflava
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...
173	False	False	False	False	False	False	False	
174	False	False	False	False	False	False	False	
175	False	False	False	False	False	False	False	
176	False	False	False	False	False	False	False	
177	False	False	False	False	False	False	False	

178 rows × 14 columns



In [58]:

```
#Обработка пропусков
data.isnull().sum()
```

Out[58]:


In [30]:

```
#При работе с первым датасетом выяснилось, что в нем нет пропусков - пробуем новый
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")

df = pd.read_csv('penguins_size.csv', sep=",")
df.head()
```

Out[30]:

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0
3	Adelie	Torgersen	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0



In [31]: `df.shape`

Out[31]:

In [32]: `df.dtypes`

Out[32]:

In [33]: `#Обработка пропусков
df.isna()`

Out[33]:

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	
0	False	False	False	False	False	False	F
1	False	False	False	False	False	False	F
2	False	False	False	False	False	False	F
3	False	False	True	True	True	True	·
4	False	False	False	False	False	False	F
...	
339	False	False	True	True	True	True	·
340	False	False	False	False	False	False	F
341	False	False	False	False	False	False	F
342	False	False	False	False	False	False	F
343	False	False	False	False	False	False	F

344 rows × 7 columns



In [40]: `df.isnull().sum()`

In [42]:

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

```
In [43]: #Фильтр по колонкам с пропущенными значениями
df_num = df[num_cols]
df_num
```

```
Out[43]:
```

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
0	39.1	18.7	181.0	3750.0
1	39.5	17.4	186.0	3800.0
2	40.3	18.0	195.0	3250.0
3	NaN	NaN	NaN	NaN
4	36.7	19.3	193.0	3450.0
...
339	NaN	NaN	NaN	NaN
340	46.8	14.3	215.0	4850.0
341	50.4	15.7	222.0	5750.0
342	45.2	14.8	212.0	5200.0
343	49.9	16.1	213.0	5400.0

344 rows × 4 columns

```
In [46]: #Выберем категориальные колонки с пропущенными значениями
cat_cols = []
for col in df.columns:
    temp_null_count = df[df[col].isnull()].shape[0]
    dt = str(df[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
    print('Колонка {}. Тип данных {}. Количество пустых значений {}'.format
```

```
In [47]: # Удалим строки, которые содержат пустые значения
df_new = df.dropna(axis=0, how='any')
(df.shape, df_new.shape)

df_new.isnull().sum()
```

Out[47]:

```
In [49]: #Преобразование категориальных признаков в числовые
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
cat_temp_df = df[['sex']]
cat_temp_df.head()

imp2 = SimpleImputer(missing_values = np.nan, strategy='most_frequent')
df_imp2 = imp2.fit_transform(cat_temp_df)
df_imp2

cat_enc = pd.DataFrame({'c1':df_imp2.T[0]})
cat_enc
```

Out[49]:

	c1
0	MALE
1	FEMALE
2	FEMALE
3	MALE
4	FEMALE
...	...
339	MALE
340	FEMALE
341	MALE
342	FEMALE
343	MALE

344 rows × 1 columns

In [51]:

```
#Кодирование категорий наборами бинарных значений
from sklearn.preprocessing import LabelEncoder, OneHotEncoder

ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc[['c1']])
cat_enc.shape
```

Out[51]:

In [52]:

```
cat_enc_ohe.shape
```

Out[52]:

In [53]:

```
cat_enc_ohe
```



```
In [54]: cat_enc_ohe.todense()[0:10]
```

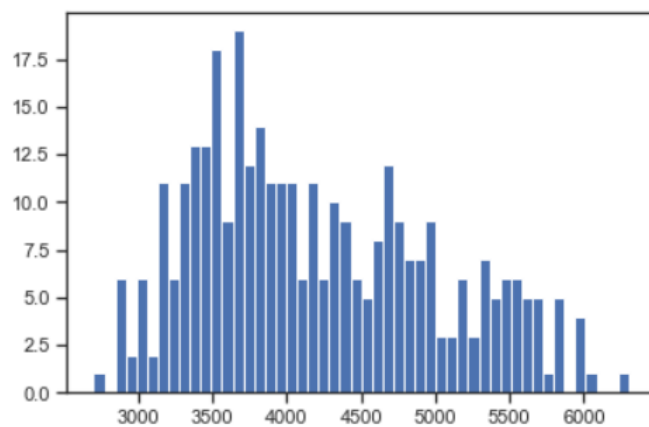
Out[54]:

```
In [55]: cat_enc.head(10)
```

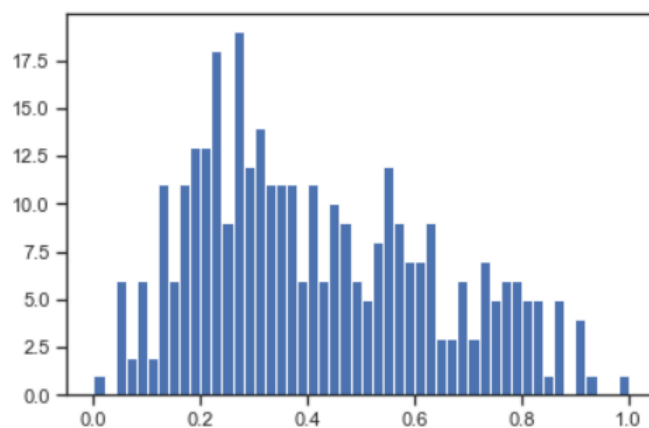
Out[55]:

	c1
0	MALE
1	FEMALE
2	FEMALE
3	MALE
4	FEMALE
5	MALE
6	FEMALE
7	MALE
8	MALE
9	MALE

```
In [56]: #Масштабирование данных  
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer  
sc1 = MinMaxScaler()  
sc1_data = sc1.fit_transform(df[['body_mass_g']])  
  
plt.hist(df['body_mass_g'], 50)  
plt.show()
```



```
In [57]: plt.hist(sc1_data, 50)  
plt.show()
```



Вывод

В ходе выполнения данной лабораторной работы я повторила язык программирования Python и работу с юпитер блокнотами.