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 ИУ5-62Б

Веревкина Диана 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 (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:
    - обработку пропусков в данных;
    - кодирование категориальных признаков;
    - масштабирование данных.

Текст программы и результаты ее выполнения

Импортируем необходимые библиотеки и датасет

```
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
```

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

344 rows × 7 columns

```
df = pd.read_csv('penguins_size.csv', sep=",")
# размер набора данных
df.shape
```

```
(344, 7)
```

```
# типы колонок
df.dtypes
```

```
species               object
island                object
culmen_length_mm     float64
culmen_depth_mm      float64
flipper_length_mm    float64
body_mass_g          float64
sex                   object
dtype: object
```

Обработка пропусков

```
#поиск пропусков
df.isna()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
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	True	True	True	True	True
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
339	False	False	True	True	True	True	True
340	False	False	False	False	False	False	False
341	False	False	False	False	False	False	False
342	False	False	False	False	False	False	False
343	False	False	False	False	False	False	False

344 rows × 7 columns

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

```
species               0
island                0
culmen_length_mm      2
culmen_depth_mm       2
flipper_length_mm     2
body_mass_g           2
sex                  10
dtype: int64
```

Обработка пропусков в числовых данных

```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
for col in df.columns:
    # Количество пустых значений
    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)
        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 culmen_length_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка culmen_depth_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка flipper_length_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка body_mass_g. Тип данных float64. Количество пустых значений 2, 0.58%.
```

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

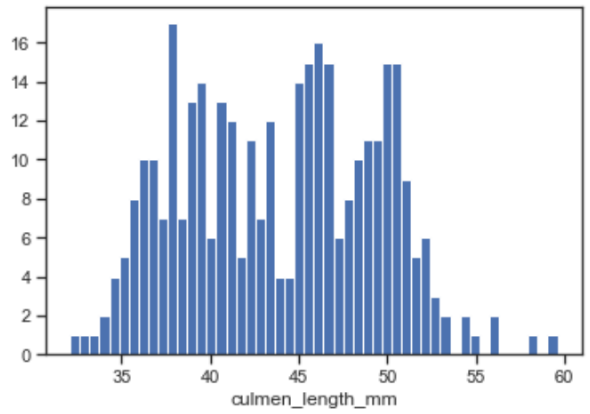
```
Колонка culmen_length_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка culmen_depth_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка flipper_length_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка body_mass_g. Тип данных float64. Количество пустых значений 2, 0.58%.
```

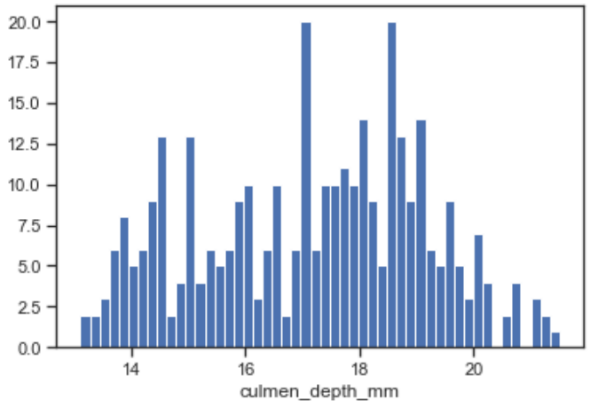
	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
0	39.1	18.7	181.0	3750.0
1	39.5	17.4	186.0	3800.0
2	40.3	18.0	195.0	3250.0
3	NaN	NaN	NaN	NaN
4	36.7	19.3	193.0	3450.0
...	...	...	...	...
339	NaN	NaN	NaN	NaN
340	46.8	14.3	215.0	4850.0
341	50.4	15.7	222.0	5750.0
342	45.2	14.8	212.0	5200.0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

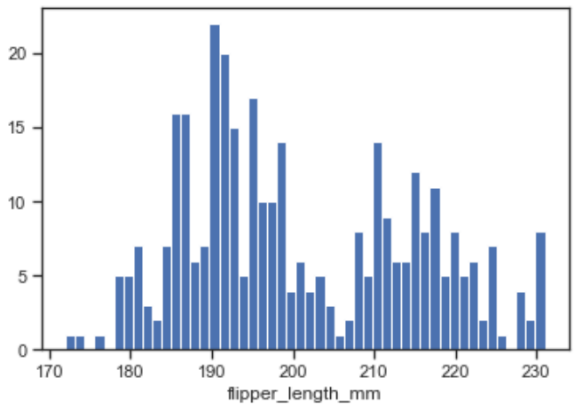
344 rows × 4 columns

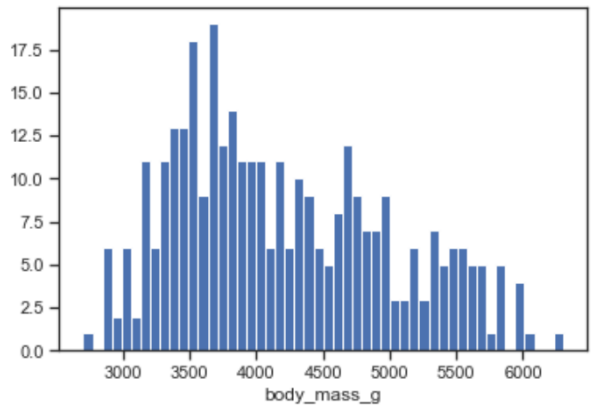
```
# Гистограмма по признакам
for col in df_num:
    plt.hist(df[col], 50)
    plt.xlabel(col)
    plt.show()
```

```
Колонка culmen_length_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка culmen_depth_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка flipper_length_mm. Тип данных float64. Количество пустых значений 2, 0.58%.
Колонка body_mass_g. Тип данных float64. Количество пустых значений 2, 0.58%.
```










```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
cat_cols = []
for col in df.columns:
    # Количество пустых значений
    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)
        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 sex. Тип данных object. Количество пустых значений 10, 2.91%.
```

Удалим строки, содержащие нулевые значения.

```
# Удаление строк, содержащих пустые значения
df_new = df.dropna(axis=0, how='any')
(df.shape, df_new.shape)

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

```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

Преобразование категориальных признаков в числовые

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
cat_temp_df = df[['sex']]
cat_temp_df.head()

# Импьютация наиболее частыми значениями
imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
df_imp2 = imp2.fit_transform(cat_temp_df)
df_imp2

#Преобразование категориальных признаков в числовые
cat_enc = pd.DataFrame({'c1':df_imp2.T[0]})
cat_enc
```

	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

Кодирование категорий наборами бинарных значений - one-hot encoding

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

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

```
(344, 1)
```

```
cat_enc_ohe.shape
```

```
(344, 3)
```

```
cat_enc_ohe
```

```
<344x3 sparse matrix of type '<class 'numpy.float64'>'
        with 344 stored elements in Compressed Sparse Row format>
```

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

```
matrix([[0., 0., 1.],
        [0., 1., 0.],
        [0., 1., 0.],
        [0., 0., 1.],
        [0., 1., 0.],
        [0., 0., 1.],
        [0., 1., 0.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.]])
```

```
cat_enc.head(10)
```

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

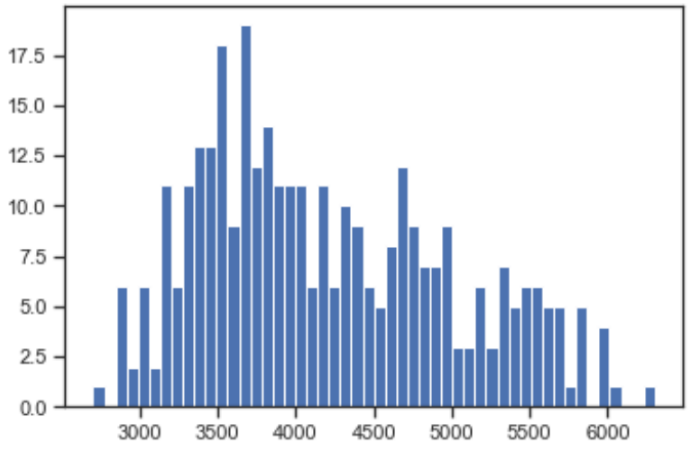
Масштабирование данных

MinMax масштабирование

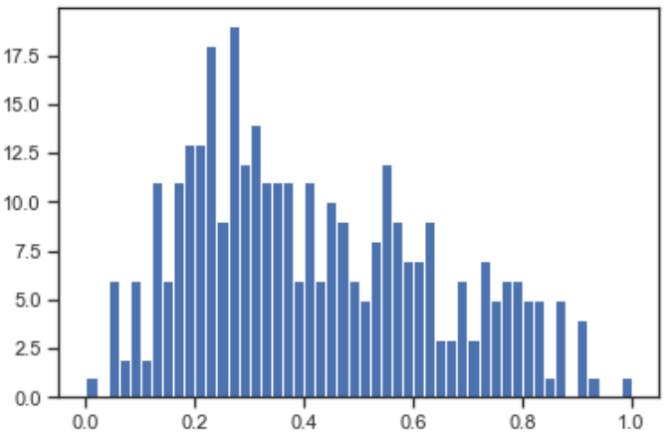
```
#Масштабирование данных
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer

sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(df[['body_mass_g']])

plt.hist(df['body_mass_g'], 50)
plt.show()
```




```
plt.hist(sc1_data, 50)
plt.show()
```



Вывод

В ходе выполнения данной лабораторной работы я повторила язык программирования Python и работу с юпитер тетрадками. Также изучила функции для обработки данных для дальнейшего формирования моделей.