

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет «Информатика и системы управления»  
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Рубежный контроль № 1  
по дисциплине «Методы машинного обучения»

**Т е м а : « М е т о д ы о б р а б о т к и  
т е к с т о в . . »**

ИСПОЛНИТЕЛЬ:

Ли Лююй

ФИО

группа ИУ5И-21М

подпись"\_\_

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю .Е

---

Москва - 2024

Группа номер	3
ЗАДАНИЕ	3+15=18    18+20=38

## Задание 18

Для набора данных проведите масштабирование данных для одного (произвольного) числового признака на основе Z-оценки.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler

np.random.seed(0)
data = {
    'feature1': np.random.randint(1, 100, 100),
    'feature2': np.random.randint(1, 100, 100),
    'feature3': np.random.rand(100) * 100
}
df = pd.DataFrame(data)

feature_to_scale = 'feature1'

scaler = StandardScaler()
df[feature_to_scale + '_scaled'] = scaler.fit_transform(df[[feature_to_scale]])

print(df.head())
```

```
   feature1  feature2  feature3  feature1_scaled
0         45         36  38.346223        -0.106026
1         48         65  74.916984         0.005580
2         65         96  87.845219         0.638015
3         68         70  10.286336         0.749621
4         68         95   9.237389         0.749621
PS F:\研究生\作业和实验\研一下\MMO\实验四\courses_current\code\rl\examples>
```

## Задание 38

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectKBest для 10 лучших признаков, и метод, основанный на взаимной информации.

```

import pandas as pd
import numpy as np
from sklearn.feature_selection import SelectKBest, mutual_info_regression

np.random.seed(0)
data = {
    'feature1': np.random.randint(1, 100, 100),
    'feature2': np.random.randint(1, 100, 100),
    'feature3': np.random.rand(100) * 100,
    'target': np.random.randint(0, 2, 100) # 假设有一个二分类目标变量
}
df = pd.DataFrame(data)

target_variable = 'target'

k_best_features = 2

X = df.drop(columns=[target_variable])

y = df[target_variable]

selector = SelectKBest(score_func=mutual_info_regression, k=k_best_features)
selected_features = selector.fit_transform(X, y)

selected_indices = selector.get_support(indices=True)

selected_feature_names = X.columns[selected_indices]

print("选定的特征:")
print(selected_feature_names)

```

```

> & C:/Users/ASUS/anaconda3/env
s/li/python.exe f:/研究生/作业和实验/研一下/MMO/实验四/courses_current/code/r1/examples/2.py
选定的特征:
Index(['feature1', 'feature3'], dtype='object')

```