Московский государственный технический университет им. Н. Э. Баумана

Курс «Технологии машинного обучения»

Отчёт по лабораторной работе №2

Выполнил:
Лупарев С. В.
группа ИУ5-63Б

Проверил:
Гапанюк Ю.Е.

Дата:

Подпись:

Дата:

Подпись:

Москва, 2025 г.

Цель лабораторной работы

Цель: 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:
   - обработку пропусков в данных;
   - кодирование категориальных признаков;
   - масштабирование данных.

Код программы и экранные формы

Набор данных - Титаник

Survived: Целевая переменная (0 = Нет, 1 = Да)

Pclass: Класс билета (1 = 1й, 2 = 2й, 3 = 3й) - категориальный

Name: Имя - текстовый

Sex: Пол - категориальный

Age: Возраст - числовой

SibSp: Количество братьев/сестер/супругов на борту - числовой

Parch: Количество родителей/детей на борту - числовой

Ticket: Номер билета - текстовый/категориальный

Fare: Стоимость проезда - числовой

Cabin: Номер каюты - категориальный

Embarked: Порт посадки (C = Шербур, Q = Квинстаун, S = Саутгемптон) - категориальный

[63]:
```
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, LabelEncoder, OrdinalEncoder
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import seaborn as sns
#df = pd.read_csv('titanic.csv')
df = sns.load_dataset('titanic')
df.head(5)
```

[63]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

[64]:

```
df_c = df.copy()
print(df_c.info())
print(df_c.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
None
```

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

[65]:
```python
# 1. Обработка пропусков
median_age = df_c['age'].median()
df_c['age'] = df_c['age'].fillna(median_age)

mode_embarked = df_c['embarked'].mode()[0]
df_c['embarked'] = df_c['embarked'].fillna(mode_embarked)

df_c = df_c.drop(['deck'], axis=1)
df_c = df_c.drop(['embark_town'], axis=1)

print(df_c.isnull().sum())
# Заменили age на медиану, embarked на моду, а deck
# просто дропнули из-за большого количества пропусков
```

```
survived      0
pclass        0
sex           0
age           0
sibsp         0
parch         0
fare          0
embarked      0
class         0
who           0
adult_male    0
alive         0
alone         0
dtype: int64
```

[66]:
```
# 2. Кодирование категориальных признаков
# преобразуем бинарные признаки
df_c['adult_male'] = df_c['adult_male'].astype(int)
df_c['alone'] = df_c['alone'].astype(int)

# преобразуем остальные категориальные признаки
categorical_cols = ['sex', 'embarked', 'who', 'alive']
df_c = pd.get_dummies(df_c, columns=categorical_cols, drop_first=True, dtype=int)

class_order = ['First', 'Second', 'Third']
ordinal_encoder = OrdinalEncoder(categories=[class_order])
encoded_class = ordinal_encoder.fit_transform(df_c[['class']])
df_c['class'] = encoded_class.astype(int)

print(df_c.head())
print(df_c.info())
```

```
   survived  pclass   age  sibsp  parch     fare  class  adult_male  alone  \
0         0       3  22.0      1      0   7.2500      2           1      0
1         1       1  38.0      1      0  71.2833      0           0      0
2         1       3  26.0      0      0   7.9250      2           0      1
3         1       1  35.0      1      0  53.1000      0           0      0
4         0       3  35.0      0      0   8.0500      2           1      1

   sex_male  embarked_Q  embarked_S  who_man  who_woman  alive_yes
0         1           0           1        1          0          0
1         0           0           0        0          1          1
2         0           0           1        0          1          1
3         0           0           1        0          1          1
4         1           0           1        1          0          0
```

```
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   survived    891 non-null    int64
 1   pclass      891 non-null    int64
 2   age         891 non-null    float64
 3   sibsp       891 non-null    int64
 4   parch       891 non-null    int64
 5   fare        891 non-null    float64
 6   class       891 non-null    int64
 7   adult_male  891 non-null    int64
 8   alone       891 non-null    int64
 9   sex_male    891 non-null    int64
 10  embarked_Q  891 non-null    int64
 11  embarked_S  891 non-null    int64
 12  who_man     891 non-null    int64
 13  who_woman   891 non-null    int64
 14  alive_yes   891 non-null    int64
dtypes: float64(2), int64(13)
memory usage: 104.5 KB
None
```

```python
features = ['pclass', 'age', 'sibsp', 'parch', 'fare', 'class']

df_c_st = df_c.copy()

scaler_standard = StandardScaler()

df_c_st[features] = scaler_standard.fit_transform(
    df_c_st[features]
)
print(df_c_st.head())
print(df_c_st[features].describe().T)
print(df_c_st.info())
```

```
   survived    pclass       age     sibsp     parch      fare     class  \
0         0  0.827377 -0.565736  0.432793 -0.473674 -0.502445  0.827377
1         1 -1.566107  0.663861  0.432793 -0.473674  0.786845 -1.566107
2         1  0.827377 -0.258337 -0.474545 -0.473674 -0.488854  0.827377
3         1 -1.566107  0.433312  0.432793 -0.473674  0.420730 -1.566107
4         0  0.827377  0.433312 -0.474545 -0.473674 -0.486337  0.827377
```

```
   adult_male  alone  sex_male  embarked_Q  embarked_S  who_man  who_woman  \
0           1      0         1           0           1        1          0
1           0      0         0           0           0        0          1
2           0      1         0           0           1        0          1
3           0      0         0           0           1        0          1
4           1      1         1           0           1        1          0

   alive_yes
0          0
1          1
2          1
3          1
4          0
         count          mean       std       min       25%       50%       75%  \
pclass   891.0 -8.772133e-17  1.000562 -1.566107 -0.369365  0.827377  0.827377
age      891.0  2.272780e-16  1.000562 -2.224156 -0.565736 -0.104637  0.433312
sibsp    891.0  4.386066e-17  1.000562 -0.474545 -0.474545 -0.474545  0.432793
parch    891.0  5.382900e-17  1.000562 -0.473674 -0.473674 -0.473674 -0.473674
fare     891.0  3.987333e-18  1.000562 -0.648422 -0.489148 -0.357391 -0.024246
class    891.0 -8.772133e-17  1.000562 -1.566107 -0.369365  0.827377  0.827377

              max
pclass   0.827377
age      3.891554
sibsp    6.784163
parch    6.974147
fare     9.667167
class    0.827377
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
```

```
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    float64
 2   age          891 non-null    float64
 3   sibsp        891 non-null    float64
 4   parch        891 non-null    float64
 5   fare         891 non-null    float64
 6   class        891 non-null    float64
 7   adult_male   891 non-null    int64
 8   alone        891 non-null    int64
 9   sex_male     891 non-null    int64
 10  embarked_Q   891 non-null    int64
 11  embarked_S   891 non-null    int64
 12  who_man      891 non-null    int64
 13  who_woman    891 non-null    int64
 14  alive_yes    891 non-null    int64
dtypes: float64(6), int64(9)
memory usage: 104.5 KB
None
```

```python
df_c_mm = df_c.copy()
scaler_minmax = MinMaxScaler()

df_c_mm[features] = scaler_minmax.fit_transform(
    df_c_mm[features]
)

print(df_c_mm.head())
print(df_c_mm[features].describe().T)
print(df_c_mm.info())
```

```
   survived  pclass       age  sibsp  parch      fare  class  adult_male  \
0         0     1.0  0.271174  0.125    0.0  0.014151    1.0           1
1         1     0.0  0.472229  0.125    0.0  0.139136    0.0           0
2         1     1.0  0.321438  0.000    0.0  0.015469    1.0           0
3         1     0.0  0.434531  0.125    0.0  0.103644    0.0           0
4         0     1.0  0.434531  0.000    0.0  0.015713    1.0           1

   alone  sex_male  embarked_Q  embarked_S  who_man  who_woman  alive_yes
0      0         1           0           1        1          0          0
1      0         0           0           0        0          1          1
2      1         0           0           1        0          1          1
3      0         0           0           1        0          1          1
4      1         1           0           1        1          0          0
        count      mean       std  min       25%       50%       75%  max
pclass  891.0  0.654321  0.418036  0.0  0.500000  1.000000  1.000000  1.0
age     891.0  0.363679  0.163605  0.0  0.271174  0.346569  0.434531  1.0
sibsp   891.0  0.065376  0.137843  0.0  0.000000  0.000000  0.125000  1.0
parch   891.0  0.063599  0.134343  0.0  0.000000  0.000000  0.000000  1.0
fare    891.0  0.062858  0.096995  0.0  0.015440  0.028213  0.060508  1.0
class   891.0  0.654321  0.418036  0.0  0.500000  1.000000  1.000000  1.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   survived    891 non-null    int64
 1   pclass      891 non-null    float64
 2   age         891 non-null    float64
 3   sibsp       891 non-null    float64
 4   parch       891 non-null    float64
 5   fare        891 non-null    float64
 6   class       891 non-null    float64
 7   adult_male  891 non-null    int64
 8   alone       891 non-null    int64
 9   sex_male    891 non-null    int64
 10  embarked_Q  891 non-null    int64
 11  embarked_S  891 non-null    int64
 12  who_man     891 non-null    int64
 13  who_woman   891 non-null    int64
 14  alive_yes   891 non-null    int64
dtypes: float64(6), int64(9)
memory usage: 104.5 KB
None
```