МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

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

ОТЧЕТ

Лабораторная работа № 2
по дисциплине «Разработка нейросетевых систем»

Тема: «Обработка признаков, часть 1»

ИСПОЛНИТЕЛЬ: ___Кравцов А.Н.__
ФИО
группа ИУ5-24М _____
подпись
"26" апреля 2024 г.

ПРЕПОДАВАТЕЛЬ: ____Гапанюк Ю.Е.____
ФИО
_____
подпись
"__"_____2024 г.

Москва – 2024

Задание

1. Выбрать набор данных (датасет), содержащий категориальные и числовые 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.) Просьба не использовать датасет, на котором данная задача решалась в лекции. 2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи: i. устранение пропусков в данных; ii. кодирование категориальных признаков; iii. нормализация числовых признаков.

Загрузка датасета

```
hdata_loaded  = pd.read_csv("netflix_titles_data.csv")
print(hdata_loaded)
```

```
      show_id     type                  title         director  \
0          s1    Movie   Dick Johnson Is Dead  Kirsten Johnson
1          s2  TV Show           Blood & Water              NaN
2          s3  TV Show               Ganglands  Julien Leclercq
3          s4  TV Show  Jailbirds New Orleans              NaN
4          s5  TV Show           Kota Factory              NaN
...       ...      ...                   ...              ...
8802    s8803    Movie                 Zodiac    David Fincher
8803    s8804  TV Show              Zombie Dumb              NaN
8804    s8805    Movie               Zombieland  Ruben Fleischer
8805    s8806    Movie                    Zoom     Peter Hewitt
8806    s8807    Movie                  Zubaan      Mozez Singh

                                                 cast        country  \
0                                                 NaN  United States
1     Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...   South Africa
2     Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...            NaN
3                                                 NaN            NaN
4     Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...          India
...                                                 ...            ...
8802  Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...  United States
8803                                              NaN            NaN
8804  Jesse Eisenberg, Woody Harrelson, Emma Stone, ...  United States
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...  United States
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...          India
...
8805  Dragged from civilian life, a former superhero...
8806  A scrappy but poor boy worms his way into a ty...
```

Устранение пропусков в данных

```
hdata = hdata_loaded
list(zip(hdata.columns, [i for i in hdata.dtypes]))
```

```
[('show_id', dtype('O')),
 ('type', dtype('O')),
 ('title', dtype('O')),
 ('director', dtype('O')),
 ('cast', dtype('O')),
 ('country', dtype('O')),
 ('date_added', dtype('O')),
 ('release_year', dtype('int64')),
 ('rating', dtype('O')),
 ('duration', dtype('O')),
 ('listed_in', dtype('O')),
 ('description', dtype('O'))]
```

```
# cols with missing values
hcols_with_na = [c for c in hdata.columns if hdata[c].isnull().sum() > 0]
hcols_with_na
```

```
['director', 'cast', 'country', 'date_added', 'rating', 'duration']
```

```
[36]
    # count
    [(c, hdata[c].isnull().sum()) for c in hcols_with_na]
```

```
[('director', 2634),
 ('cast', 825),
 ('country', 831),
 ('date_added', 10),
 ('rating', 4),
 ('duration', 3)]
```

```
[42]
    # percent
    [(c, hdata[c].isnull().mean()) for c in hcols_with_na]
```

```
[('director', 0.0),
 ('cast', 0.0),
 ('country', 0.0),
 ('date_added', 0.0),
 ('rating', 0.0),
 ('duration', 0.0)]
```

Заполнение показателями центра распределения и константой

```
def impute_column(dataset, column, strategy_param, fill_value_param=None):
    """
    Заполнение пропусков в одном признаке
    """
    temp_data = dataset[[column]].values
    size = temp_data.shape[0]

    indicator = MissingIndicator()
    mask_missing_values_only = indicator.fit_transform(temp_data)

    imputer = SimpleImputer(strategy=strategy_param,
                            fill_value=fill_value_param)
    all_data = imputer.fit_transform(temp_data)

    missed_data = temp_data[mask_missing_values_only]
    filled_data = all_data[mask_missing_values_only]

    return all_data.reshape((size,)), filled_data, missed_data
```

```
all_data, filled_data, missed_data = impute_column(hdata, 'director', 'constant', 'unknown')
all_data, filled_data, missed_data
```

```
(array(['Kirsten Johnson', 'unknown', 'Julien Leclercq', ...,
        'Ruben Fleischer', 'Peter Hewitt', 'Mozez Singh'], dtype=object),
 array(['unknown', 'unknown', 'unknown', ..., 'unknown', 'unknown',
        'unknown'], dtype=object),
 array([nan, nan, nan, ..., nan, nan, nan], dtype=object))
```

```
hcols_with_na
```

[40]

```
['director', 'cast', 'country', 'date_added', 'rating', 'duration']
```

```
for i in hcols_with_na:
    if i == 'rating' or i == 'duration':
        all_data, filled_data, missed_data = impute_column(hdata, i, 'most_frequent')
    else:
        all_data, filled_data, missed_data = impute_column(hdata, i, 'constant', 'unknown')
    hdata[i] = all_data
hdata.isnull().sum()
```

[41]

```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

```
hcols_with_na = [c for c in hdata_loaded.columns if hdata_loaded[c].isnull().sum() > 0]
[(c, hdata_loaded[c].isnull().sum()) for c in hcols_with_na]
```

6]

```
[]
```

```
res = hdata_loaded.dropna(axis=0, how='any')
```

7]

+ Code + Markdown

```
hcols_with_na = [c for c in res.columns if res[c].isnull().sum() > 0]
[(c, res[c].isnull().sum()) for c in hcols_with_na]
```

8]

```
[]
```

Кодирование категориальных признаков

```
res
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	unknown	United States	September 25, 2021	2020	PG-13	90 min	Documer
1	s2	TV Show	Blood & Water	unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Sho Dramas, TV My
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	unknown	September 24, 2021	2021	TV-MA	1 Season	Crime TV S International TV Sho
3	s4	TV Show	Jailbirds New Orleans	unknown	unknown	unknown	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Rea
4	s5	TV Show	Kota Factory	unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV S Romantic TV Shows
...	...	...	...	...	...	...	...	...	...	...	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert	United States	November 20, 2019	2007	R	158 min	Cult Movies, D

```
#  Count encoding предполагает что значение категории заменяется на количество раз, которое оно встречается в категории.
from category_encoders.count import CountEncoder as ce_CountEncoder
ce_CountEncoder1 = ce_CountEncoder()
data_COUNT_ENC = ce_CountEncoder1.fit_transform(res[res.columns.difference(['director'])])
data_COUNT_ENC
```

	cast	country	date_added	description	duration	listed_in	rating	release_year	show_id	title	type
0	825	2818	1	1	152	359	490	2020	1	1	6131
1	1	30	10	1	425	26	3211	2021	1	1	2676
2	1	831	10	1	1796	18	3211	2021	1	1	2676
3	825	831	10	1	1796	16	3211	2021	1	1	2676
4	1	972	10	1	425	94	3211	2021	1	1	2676
...	...	...	...	...	...	...	...	...	...	...	...
8802	1	2818	30	1	12	1	799	2007	1	1	6131
8803	825	831	52	1	425	4	334	2018	1	1	2676
8804	1	2818	89	1	116	12	799	2009	1	1	6131
8805	1	2818	1	1	116	201	287	2006	1	1	6131
8806	1	972	10	1	68	57	2160	2015	1	1	6131

8807 rows × 11 columns

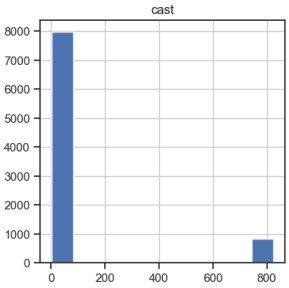
Нормализация числовых признаков

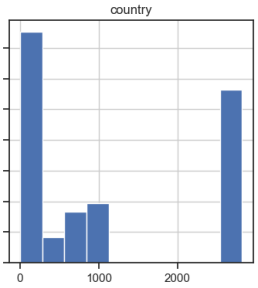
Сохранение для следующей работы

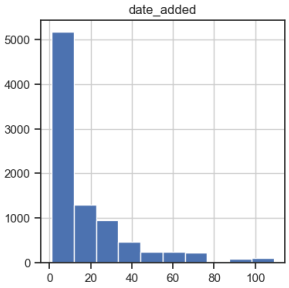
```
res = data_COUNT_ENC
res
```

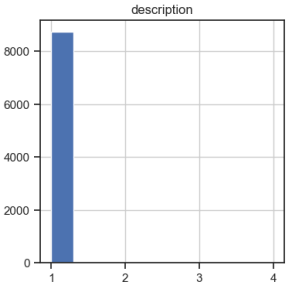
	cast	country	date_added	description	duration	listed_in	rating	release_year	show_id	title	type
0	825	2818	1	1	152	359	490	2020	1	1	6131
1	1	30	10	1	425	26	3211	2021	1	1	2676
2	1	831	10	1	1796	18	3211	2021	1	1	2676
3	825	831	10	1	1796	16	3211	2021	1	1	2676
4	1	972	10	1	425	94	3211	2021	1	1	2676
...	...	...	...	...	...	...	...	...	...	...	...
8802	1	2818	30	1	12	1	799	2007	1	1	6131
8803	825	831	52	1	425	4	334	2018	1	1	2676
8804	1	2818	89	1	116	12	799	2009	1	1	6131
8805	1	2818	1	1	116	201	287	2006	1	1	6131
8806	1	972	10	1	68	57	2160	2015	1	1	6131

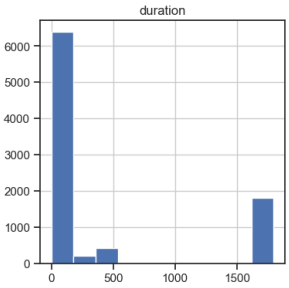
8807 rows × 11 columns

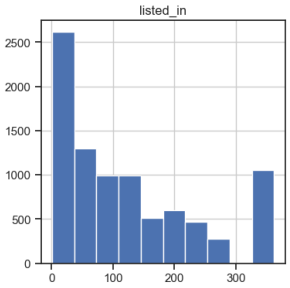

cast

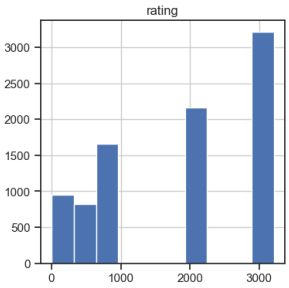

country

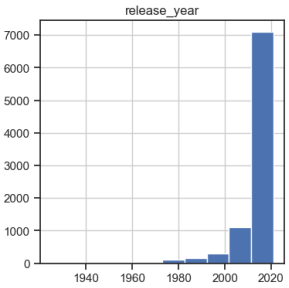

date_added

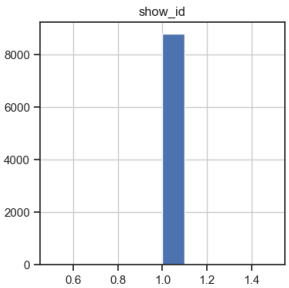

description

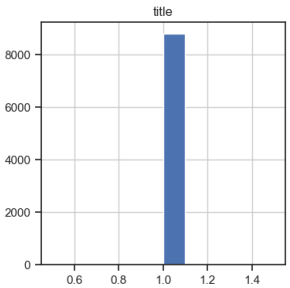

duration

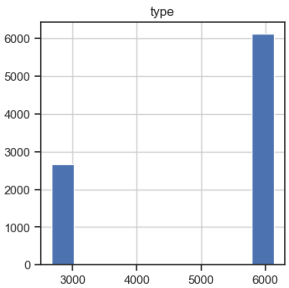

listed_in

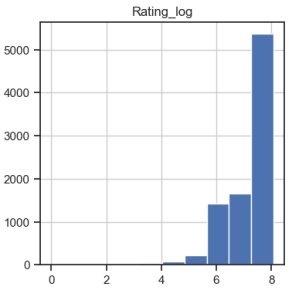

rating

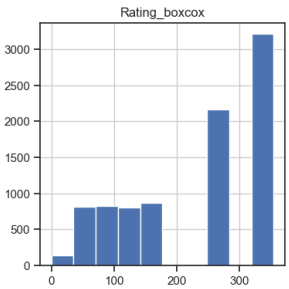

release_year

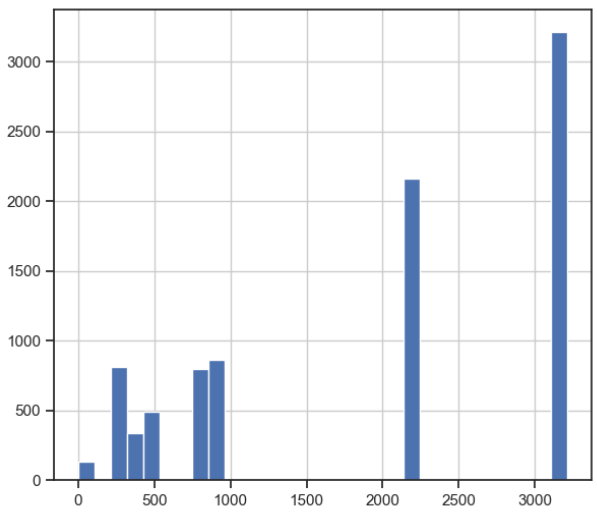

show_id

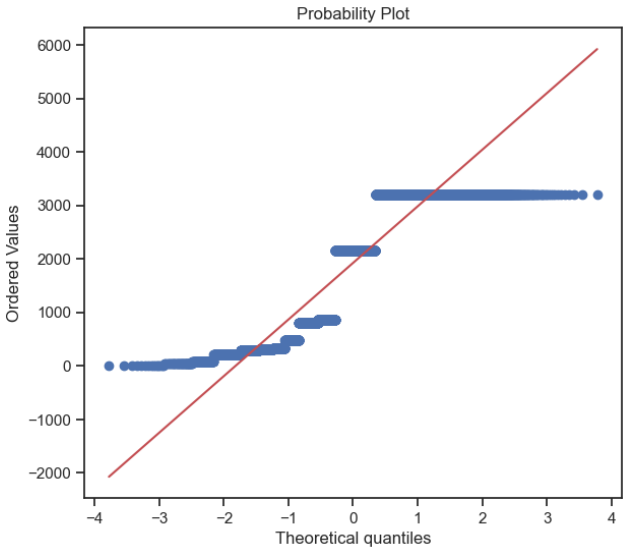

title


type

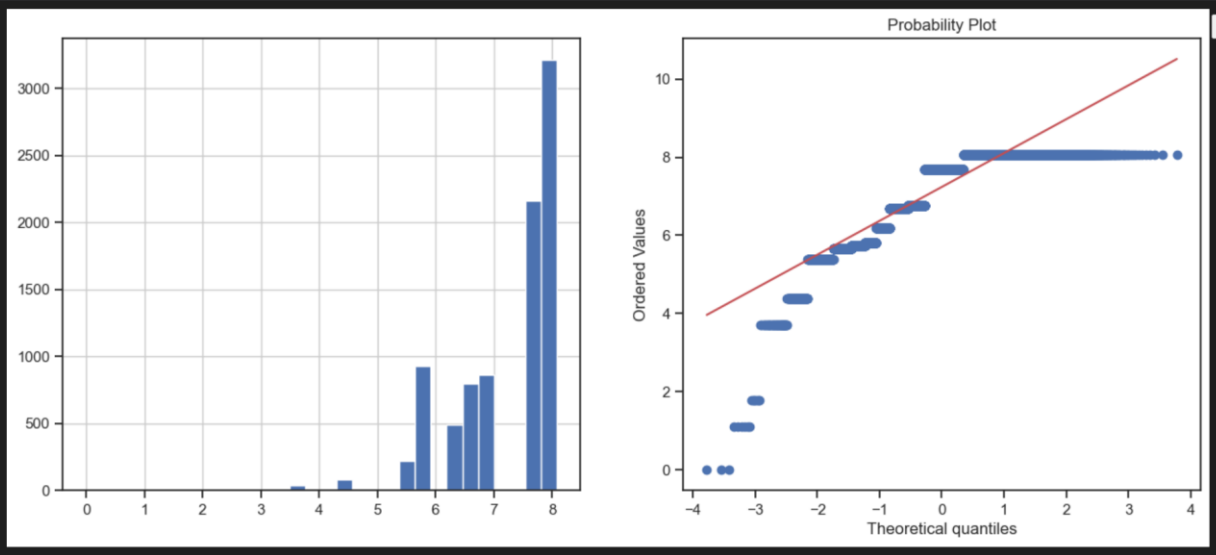

Rating_log


Rating_boxcox




Probability Plot
Ordered Values
Theoretical quantiles

```
#  Логарифмическое преобразование
res['Rating_log'] = np.log(res['rating'])
diagnostic_plots(res, 'Rating_log')
```




```
# Преобразование Бокса-Кокса
res['Rating_boxcox'], param = stats.boxcox(res['rating'])
print('Оптимальное значение λ = {}'.format(param))
diagnostic_plots(res, 'Rating_boxcox')
```

```
Оптимальное значение λ = 0.6799528147632296
```