Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №3
по дисциплине

«Методы машинного обучения»

на тему

«Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных»

Выполнил:
студент группы ИУ5-61Б
Белоусов Е. А.

____

Москва — 2020 г.

1. Лабораторная работа 3

2. Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных.

2.1. Цель

изучение способов предварительной обработки данных для дальнейшего формирования моделей.

2.2. 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.

```
In [1]: import numpy as np
        import pandas as pd
        import sklearn as sk
```

2.3. Загрузка и первичный анализ данных

```
In [2]: data = pd.read_csv('fake_job_postings.csv')
```

```
In [3]: data.head()
```

```
Out[3]:   job_id                               title            location  \
     0       1                     Marketing Intern    US, NY, New York
     1       2  Customer Service - Cloud Video Production    NZ, , Auckland
     2       3    Commissioning Machinery Assistant (CMA)       US, IA, Wever
     3       4          Account Executive - Washington DC  US, DC, Washington
     4       5                  Bill Review Manager  US, FL, Fort Worth

      department salary_range                              company_profile  \
     0  Marketing          NaN  We're Food52, and we've created a groundbreaki…
     1    Success          NaN  90 Seconds, the worlds Cloud Video Production …
     2        NaN          NaN  Valor Services provides Workforce Solutions th…
     3      Sales          NaN  Our passion for improving quality of life thro…
     4        NaN          NaN  SpotSource Solutions LLC is a Global Human Cap…

                                     description  \
     0  Food52, a fast-growing, James Beard Award-winn…
     1  Organised - Focused - Vibrant - Awesome!Do you…
     2  Our client, located in Houston, is actively se…
     3  THE COMPANY: ESRI – Environmental Systems Rese…
     4  JOB TITLE: Itemization Review ManagerLOCATION:…
```

```
                                        requirements  \
0  Experience with content management systems a m…
1  What we expect from you:Your key responsibilit…
2  Implement pre-commissioning and commissioning …
3  EDUCATION: Bachelor’s or Master’s in GIS, busi…
4  QUALIFICATIONS:RN license in the State of Texa…

                                            benefits  telecommuting  \
0                                                NaN              0
1  What you will get from usThrough being part of…              0
2                                                NaN              0
3  Our culture is anything but corporate—we have …              0
4                              Full Benefits Offered              0

   has_company_logo  has_questions employment_type required_experience  \
0                 1              0           Other          Internship
1                 1              0       Full-time      Not Applicable
2                 1              0             NaN                 NaN
3                 1              0       Full-time    Mid-Senior level
4                 1              1       Full-time    Mid-Senior level

  required_education                   industry              function  \
0                NaN                        NaN             Marketing
1                NaN  Marketing and Advertising      Customer Service
2                NaN                        NaN                   NaN
3  Bachelor's Degree          Computer Software                 Sales
4  Bachelor's Degree     Hospital & Health Care  Health Care Provider

   fraudulent
0           0
1           0
2           0
3           0
4           0
```

```
In [4]: data.shape
```

```
Out[4]: (17880, 18)
```

```
In [6]: # проверим есть ли пропущенные значения
        data.isnull().sum()
```

```
Out[6]: job_id                     0
        title                      0
        location                 346
        department             11547
        salary_range           15012
        company_profile         3308
        description                1
        requirements            2695
```

```
benefits               7210
telecommuting             0
has_company_logo          0
has_questions             0
employment_type        3471
required_experience    7050
required_education     8105
industry               4903
function               6455
fraudulent                0
dtype: int64
```

```
In [7]: # типы колонок
        data.dtypes
```

```
Out[7]: job_id                  int64
        title                  object
        location               object
        department             object
        salary_range           object
        company_profile        object
        description            object
        requirements           object
        benefits               object
        telecommuting           int64
        has_company_logo        int64
        has_questions           int64
        employment_type        object
        required_experience    object
        required_education     object
        industry               object
        function               object
        fraudulent              int64
        dtype: object
```

2.4. 1. Обработка пропусков в данных

```
In [11]: # Удаление строк, содержащих пустые значения
         data_new = data.dropna(axis=0, how='any')
         (data.shape, data_new.shape)
```

```
Out[11]: ((17880, 18), (774, 18))
```

```
In [12]: data_new.isnull().sum()
```

```
Out[12]: job_id                 0
         title                  0
         location               0
         department             0
         salary_range           0
         company_profile        0
```

```
description           0
requirements          0
benefits              0
telecommuting         0
has_company_logo      0
has_questions         0
employment_type       0
required_experience   0
required_education    0
industry              0
function              0
fraudulent            0
dtype: int64
```

2.5. 2. Преобразование категориальных признаков в числовые

```
In [19]: from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

```
In [28]: le = LabelEncoder()
         cat_enc_le = le.fit_transform(data_new['required_education'])
         cat_enc_le
```

```
Out[28]: array([4, 1, 7, 3, 1, 1, 1, 3, 1, 1, 1, 3, 1, 1, 1, 1, 7, 1, 1, 1, 1, 7, 7,
                1, 1, 7, 3, 2, 3, 1, 4, 1, 7, 2, 7, 3, 2, 3, 1, 3, 6, 1, 1, 4, 1, 3,
                4, 1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 7, 1, 3, 7, 1, 1, 1, 3, 1, 1, 0, 1,
                3, 1, 1, 7, 7, 3, 1, 1, 1, 1, 1, 3, 3, 0, 1, 7, 1, 3, 3, 1, 1, 7, 7,
                1, 3, 3, 1, 1, 3, 7, 1, 1, 1, 3, 1, 1, 7, 1, 7, 3, 3, 1, 1, 7, 7, 1,
                1, 8, 7, 3, 3, 1, 1, 3, 7, 3, 7, 1, 1, 7, 1, 1, 1, 1, 6, 3, 7, 0, 7,
                1, 1, 1, 1, 1, 1, 3, 1, 7, 7, 7, 3, 3, 7, 3, 1, 1, 1, 3, 1, 1, 3, 7,
                7, 7, 4, 7, 1, 3, 1, 1, 1, 7, 1, 1, 3, 1, 0, 8, 7, 7, 1, 3, 1, 1, 3,
                7, 1, 7, 1, 3, 3, 3, 3, 2, 4, 1, 3, 3, 3, 3, 3, 1, 3, 3, 1, 3, 3, 1,
                3, 1, 3, 1, 1, 1, 1, 0, 1, 1, 7, 1, 7, 7, 1, 1, 1, 1, 7, 7, 1, 4, 7,
                1, 1, 7, 7, 1, 1, 4, 7, 1, 1, 1, 3, 6, 2, 1, 3, 1, 4, 7, 3, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 7, 1, 1, 7, 3, 3, 1, 7, 3, 3, 3, 3, 3, 3, 3, 3, 3,
                3, 3, 3, 1, 1, 1, 1, 1, 6, 3, 1, 1, 1, 4, 3, 1, 3, 1, 1, 1, 1, 1, 1,
                1, 4, 1, 1, 1, 1, 3, 1, 1, 7, 7, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
                3, 3, 3, 3, 3, 3, 3, 3, 4, 1, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 7, 3, 7,
                3, 3, 1, 1, 3, 3, 3, 1, 3, 3, 3, 3, 1, 3, 3, 7, 1, 0, 1, 3, 6, 1, 3,
                1, 3, 1, 1, 6, 1, 6, 1, 1, 1, 4, 1, 3, 1, 3, 3, 1, 3, 1, 1, 1, 1, 8,
                1, 0, 1, 1, 3, 7, 7, 7, 3, 3, 1, 1, 1, 7, 7, 3, 3, 7, 1, 7, 1, 7, 0,
                1, 3, 1, 1, 1, 1, 1, 3, 3, 0, 1, 0, 7, 1, 1, 6, 1, 1, 7, 3, 1, 1, 4,
                1, 1, 1, 1, 3, 1, 3, 0, 1, 3, 7, 1, 3, 1, 3, 1, 1, 1, 7, 1, 7, 0, 7,
                1, 1, 1, 1, 1, 1, 1, 8, 3, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 7, 1, 1,
                1, 1, 1, 1, 0, 0, 1, 2, 7, 1, 7, 7, 7, 0, 1, 3, 1, 3, 1, 1, 7, 1, 1,
                0, 0, 1, 7, 1, 1, 1, 7, 1, 4, 1, 1, 0, 4, 1, 1, 1, 0, 1, 3, 3, 2, 1,
                1, 3, 1, 1, 2, 2, 3, 2, 1, 7, 3, 7, 7, 1, 1, 7, 0, 9, 3, 1, 1, 2, 0,
                1, 1, 3, 5, 1, 0, 7, 0, 1, 1, 4, 3, 1, 1, 1, 3, 1, 1, 1, 7, 3, 0, 1,
                3, 3, 7, 1, 1, 7, 1, 4, 7, 1, 7, 7, 3, 1, 7, 0, 1, 0, 0, 1, 3, 1, 1,
                0, 1, 1, 1, 0, 1, 1, 1, 3, 1, 7, 7, 1, 2, 7, 7, 3, 7, 3, 1, 3, 1, 1,
                1, 7, 1, 1, 1, 1, 0, 1, 7, 4, 1, 0, 3, 1, 1, 1, 1, 1, 1, 0, 1, 3, 1,
                7, 1, 0, 1, 1, 1, 1, 2, 1, 3, 7, 7, 0, 1, 1, 3, 3, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 7, 3, 7, 1, 0, 1, 3, 7, 1, 1, 7, 7, 5, 7, 2, 3, 1, 1,
                7, 1, 1, 3, 1, 3, 8, 7, 7, 7, 1, 1, 1, 9, 3, 3, 1, 1, 1, 5, 1, 1, 7,
                1, 3, 4, 1, 1, 1, 1, 4, 7, 7, 1, 1, 1, 1, 1, 4, 1, 2, 1, 1, 1, 7, 7,
                1, 1, 3, 1, 4, 3, 1, 7, 1, 1, 1, 7, 1, 1, 6, 1, 3, 1, 1, 7, 1, 1, 4,
                1, 1, 3, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 1])
```

```
In [29]: data_new['required_education'].unique()
```

```
Out[29]: array(["Master's Degree", "Bachelor's Degree", 'Unspecified',
        'High School or equivalent', 'Certification',
        'Some College Coursework Completed', 'Associate Degree',
        'Vocational', 'Vocational - HS Diploma', 'Professional'], dtype=object)
```

```
In [30]: np.unique(cat_enc_le)
```

```
Out[30]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [31]: le.inverse_transform([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
Out[31]: array(['Associate Degree', "Bachelor's Degree", 'Certification',
        'High School or equivalent', "Master's Degree", 'Professional',
        'Some College Coursework Completed', 'Unspecified', 'Vocational',
        'Vocational - HS Diploma'], dtype=object)
```

2.6. 3. Кодирование категорий наборами бинарных значений - one-hot encoding

```
In [48]: ohe = OneHotEncoder()
    data_encoded, data_categories = data_new['required_education'].factorize()
    cat_enc_ohe = ohe.fit_transform(data_encoded.reshape(-1, 1))
    cat_enc_ohe.shape
```

```
Out[48]: (774, 10)
```

```
In [49]: data_encoded.shape
```

```
Out[49]: (774,)
```

```
In [50]: cat_enc_ohe
```

```
Out[50]: <774x10 sparse matrix of type '<type 'numpy.float64'>'
         with 774 stored elements in Compressed Sparse Row format>
```

```
In [51]: cat_enc_ohe.todense()[0:10]
```

```
Out[51]: matrix([[ 1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
         [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.]])
```

```
In [52]: data_categories

Out[52]: Index([u'Master's Degree', u'Bachelor's Degree', u'Unspecified',
            u'High School or equivalent', u'Certification',
            u'Some College Coursework Completed', u'Associate Degree',
            u'Vocational', u'Vocational - HS Diploma', u'Professional'],
           dtype='object')
```

2.7. Масштабирование данных

Для масштабирования данных будем использовать другой набор данных

```
In [54]: data = pd.read_csv('winequality-red.csv')

In [55]: data.shape

Out[55]: (1599, 12)

In [56]: data.head()

Out[56]:    fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
         0            7.4              0.70         0.00             1.9      0.076
         1            7.8              0.88         0.00             2.6      0.098
         2            7.8              0.76         0.04             2.3      0.092
         3           11.2              0.28         0.56             1.9      0.075
         4            7.4              0.70         0.00             1.9      0.076

            free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
         0                 11.0                  34.0   0.9978  3.51       0.56
         1                 25.0                  67.0   0.9968  3.20       0.68
         2                 15.0                  54.0   0.9970  3.26       0.65
         3                 17.0                  60.0   0.9980  3.16       0.58
         4                 11.0                  34.0   0.9978  3.51       0.56

            alcohol  quality
         0      9.4        5
         1      9.8        5
         2      9.8        5
         3      9.8        6
         4      9.4        5

In [57]: data.dtypes

Out[57]: fixed acidity           float64
         volatile acidity        float64
         citric acid             float64
         residual sugar          float64
         chlorides               float64
         free sulfur dioxide     float64
         total sulfur dioxide    float64
         density                 float64
         pH                      float64
```

```
          sulphates             float64
          alcohol              float64
          quality               int64
          dtype: object
```

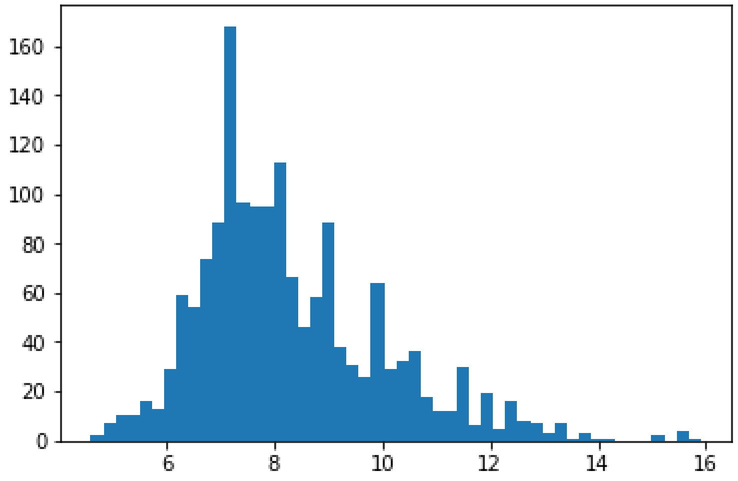
```
In [58]: data.isnull().sum()
```

```
Out[58]: fixed acidity          0
          volatile acidity       0
          citric acid            0
          residual sugar         0
          chlorides              0
          free sulfur dioxide    0
          total sulfur dioxide   0
          density                0
          pH                     0
          sulphates              0
          alcohol                0
          quality                0
          dtype: int64
```

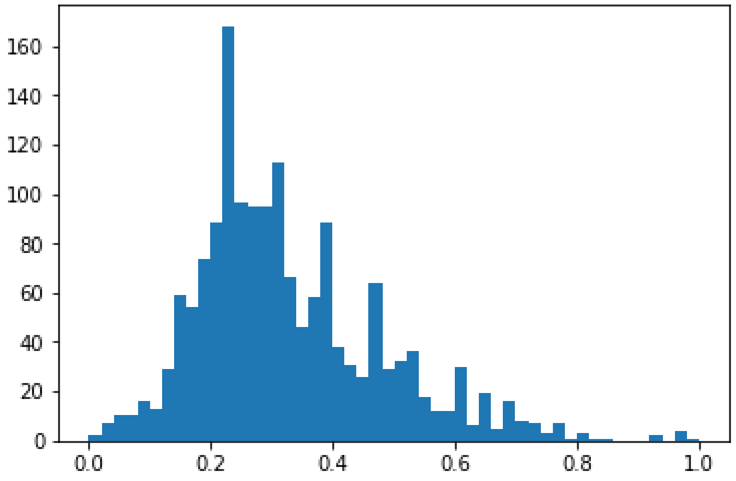
```
In [59]: from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

```
In [60]: sc1 = MinMaxScaler()
          sc1_data = sc1.fit_transform(data[['fixed acidity']])
```

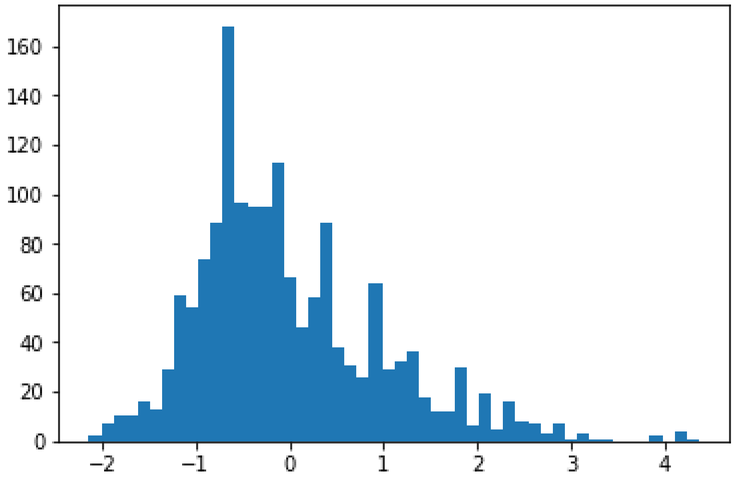
```
In [62]: import matplotlib.pyplot as plt
          %matplotlib inline
          plt.hist(data['fixed acidity'], 50)
          plt.show()
```



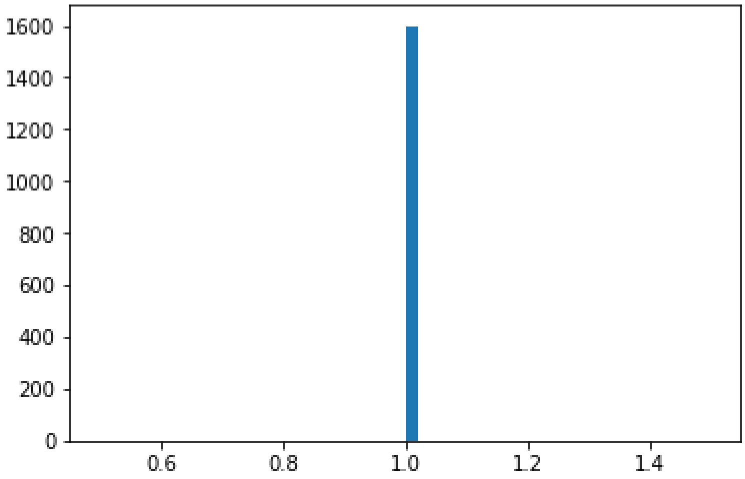
```
In [63]: plt.hist(sc1_data, 50)
          plt.show()
```



```
In [65]: sc2 = StandardScaler()
         sc2_data = sc2.fit_transform(data[['fixed acidity']])
         plt.hist(sc2_data, 50)
         plt.show()
```



```
In [67]: sc3 = Normalizer()
         sc3_data = sc3.fit_transform(data[['fixed acidity']])
         plt.hist(sc3_data, 50)
         plt.show()
```


1600
1400
1200
1000
800
600
400
200
0
0.6
0.8
1.0
1.2
1.4