Московский государственный технический

университет им. Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №3

Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных

Группа: ИУ5-62Б

Студент: Селедкина А.С.

Преподаватель: Гапанюк Ю.Е.

Москва, 2020 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 примеры выполнения

```
import numpy as np
import pandas as pd
pd.set_option('display.max.columns', 100)
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
from sklearn.datasets import *
data = pd.read_csv('data/HRDataset_v13.csv')
data.shape
```

```
Out[3]: (401, 35)
```

```
data.head()
```

	Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversityJobFairID	PayRate	Termd	Positic
0	Brown, Mia	1.103024e+09	1.0	1.0	0.0	1.0	1.0	3.0	1.0	28.50	0.0	
1	LaRotonda, William	1.106027e+09	0.0	2.0	1.0	1.0	1.0	3.0	0.0	23.00	0.0	
2	Steans, Tyrone	1.302053e+09	0.0	0.0	1.0	1.0	1.0	3.0	0.0	29.00	0.0	
3	Howard, Estelle	1.211051e+09	1.0	1.0	0.0	1.0	1.0	3.0	0.0	21.50	1.0	
4	Singh, Nan	1.307060e+09	0.0	0.0	0.0	1.0	1.0	3.0	0.0	16.56	0.0	

Обработка пропусков в данных

```
data.isnull().sum()
```

Out[5]:
```
Employee_Name                91
EmpID                        91
MarriedID                    91
MaritalStatusID              91
GenderID                     91
EmpStatusID                  91
DeptID                       91
PerfScoreID                  91
FromDiversityJobFairID       91
PayRate                      91
Termd                        91
PositionID                   91
Position                     91
State                        91
Zip                          91
DOB                          91
Sex                          91
MaritalDesc                  91
CitizenDesc                  91
HispanicLatino               91
RaceDesc                     91
DateofHire                   91
DateofTermination           298
TermReason                   92
EmploymentStatus             91
Department                   91
ManagerName                  91
ManagerID                    99
RecruitmentSource            91
PerformanceScore             91
EngagementSurvey             91
EmpSatisfaction              91
SpecialProjectsCount         91
LastPerformanceReview_Date  194
DaysLateLast30              194
dtype: int64
```

```
data1 = data.dropna(axis=0, how='all')
data1.isnull().sum()
```

```
Out[7]: Employee_Name                0
        EmpID                        0
        MarriedID                    0
        MaritalStatusID              0
        GenderID                     0
        EmpStatusID                  0
        DeptID                       0
        PerfScoreID                  0
        FromDiversityJobFairID       0
        PayRate                      0
        Termd                        0
        PositionID                   0
        Position                     0
        State                        0
        Zip                          0
        DOB                          0
        Sex                          0
        MaritalDesc                  0
        CitizenDesc                  0
        HispanicLatino               0
        RaceDesc                     0
        DateofHire                   0
        DateofTermination          207
        TermReason                   1
        EmploymentStatus             0
        Department                   0
        ManagerName                  0
        ManagerID                    8
        RecruitmentSource            0
        PerformanceScore             0
        EngagementSurvey             0
        EmpSatisfaction              0
        SpecialProjectsCount         0
        LastPerformanceReview_Date   103
        DaysLateLast30               103
        dtype: int64
```

```
data1.shape
```

```
Out[8]: (310, 35)
```

```
data1['TermReason'].dtype
```

```
Out[9]: dtype('O')
```

```
data1['TermReason'].unique()
```

```
Out[10]: array(['N/A - still employed', nan, 'career change', 'Another position',
                'attendance', 'relocation out of area',
                'N/A - Has not started yet', 'performance', 'no-call, no-show',
                'hours', 'medical issues', 'retiring', 'unhappy', 'more money',
                'return to school', 'gross misconduct', 'military',
                'maternity leave - did not return'], dtype=object)
```

```
from sklearn.impute import SimpleImputer
imp = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp = imp.fit(data1[['TermReason']])
data1.isnull().sum()
```

Out[12]:
```
Employee_Name                  0
EmpID                          0
MarriedID                      0
MaritalStatusID                0
GenderID                       0
EmpStatusID                    0
DeptID                         0
PerfScoreID                    0
FromDiversityJobFairID         0
PayRate                        0
Termd                          0
PositionID                     0
Position                       0
State                          0
Zip                            0
DOB                            0
Sex                            0
MaritalDesc                    0
CitizenDesc                    0
HispanicLatino                 0
RaceDesc                       0
DateofHire                     0
DateofTermination            207
TermReason                     1
EmploymentStatus               0
Department                     0
ManagerName                    0
ManagerID                      8
RecruitmentSource              0
PerformanceScore               0
EngagementSurvey               0
EmpSatisfaction                0
SpecialProjectsCount           0
LastPerformanceReview_Date   103
DaysLateLast30               103
dtype: int64
```

```
data2 = data1.dropna(axis=1, how='any')
data2.isnull().sum()
```

```
Out[14]: Employee_Name              0
         EmpID                      0
         MarriedID                  0
         MaritalStatusID            0
         GenderID                   0
         EmpStatusID                0
         DeptID                     0
         PerfScoreID                0
         FromDiversityJobFairID     0
         PayRate                    0
         Termd                      0
         PositionID                 0
         Position                   0
         State                      0
         Zip                        0
         DOB                        0
         Sex                        0
         MaritalDesc                0
         CitizenDesc                0
         HispanicLatino             0
         RaceDesc                   0
         DateofHire                 0
         EmploymentStatus           0
         Department                 0
         ManagerName                0
         RecruitmentSource          0
         PerformanceScore           0
         EngagementSurvey           0
         EmpSatisfaction            0
         SpecialProjectsCount       0
         dtype: int64
```

Кодирование категориальных признаков

```
data2.head()
```

	Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversityJobFairID	PayRate	Termd	Positic
0	Brown, Mia	1.103024e+09	1.0	1.0	0.0	1.0	1.0	3.0	1.0	28.50	0.0	
1	LaRotonda, William	1.106027e+09	0.0	2.0	1.0	1.0	1.0	3.0	0.0	23.00	0.0	
2	Steans, Tyrone	1.302053e+09	0.0	0.0	1.0	1.0	1.0	3.0	0.0	29.00	0.0	
3	Howard, Estelle	1.211051e+09	1.0	1.0	0.0	1.0	1.0	3.0	0.0	21.50	1.0	
4	Singh, Nan	1.307060e+09	0.0	0.0	0.0	1.0	1.0	3.0	0.0	16.56	0.0	

```
for col in data2.columns:
    dt = str(data2[col].dtype)
    if dt=='object' and len(data2[col].unique()) < 10:
        print('Колонка {}. Тип данных {}. Уникальные
            значения:\n{}\n'.format(col, dt, data2[col].unique()))
```

```
Колонка Sex. Тип данных object. Уникальные значения:
['F' 'M ']

Колонка MaritalDesc. Тип данных object. Уникальные значения:
['Married' 'Divorced' 'Single' 'Widowed' 'Separated']

Колонка CitizenDesc. Тип данных object. Уникальные значения:
['US Citizen' 'Eligible NonCitizen' 'Non-Citizen']

Колонка HispanicLatino. Тип данных object. Уникальные значения:
['No' 'Yes' 'yes' 'no']

Колонка RaceDesc. Тип данных object. Уникальные значения:
['Black or African American' 'White' 'Asian'
 'American Indian or Alaska Native' 'Two or more races' 'Hispanic']

Колонка EmploymentStatus. Тип данных object. Уникальные значения:
['Active' 'Terminated for Cause' 'Voluntarily Terminated' 'Future Start'
 'Leave of Absence']

Колонка Department. Тип данных object. Уникальные значения:
['Admin Offices' 'Sales' 'IT/IS' 'Production       ' 'Executive Office'
 'Software Engineering']

Колонка PerformanceScore. Тип данных object. Уникальные значения:
['Fully Meets' 'PIP' 'Exceeds' 'Needs Improvement']
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data_le = le.fit_transform(data2['MaritalDesc'])
np.unique(data_le)
```

```
Out[18]: array([0, 1, 2, 3, 4])
```

```
data_le
```

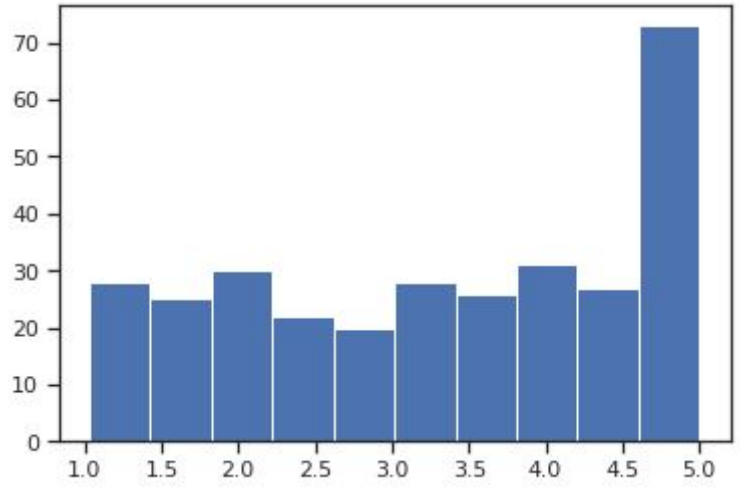
```
Out[19]: array([1, 0, 3, 1, 3, 1, 1, 3, 3, 1, 3, 4, 1, 3, 2, 3, 3, 1, 1, 3, 1, 3,
                3, 3, 1, 3, 3, 1, 3, 3, 1, 2, 3, 1, 3, 1, 1, 1, 3, 3, 3, 3, 1, 3,
                1, 0, 1, 1, 1, 1, 0, 1, 3, 3, 1, 3, 3, 0, 1, 3, 3, 3, 3, 1, 0, 1,
                1, 2, 3, 1, 1, 1, 1, 1, 0, 3, 3, 3, 0, 1, 1, 1, 3, 1, 1, 0, 1, 1,
                3, 3, 3, 1, 1, 3, 1, 1, 4, 1, 3, 1, 1, 3, 1, 3, 1, 1, 1, 1, 0, 3,
                1, 1, 2, 1, 4, 1, 0, 3, 1, 1, 3, 3, 3, 1, 3, 1, 1, 3, 3, 2, 0, 3,
                4, 1, 3, 3, 1, 3, 0, 1, 1, 1, 0, 1, 0, 1, 3, 3, 3, 3, 1, 3, 1, 3,
                0, 2, 3, 3, 1, 3, 3, 1, 1, 4, 0, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 3,
                0, 3, 1, 2, 3, 1, 3, 1, 0, 3, 1, 3, 0, 0, 3, 0, 0, 3, 0, 3, 3, 3,
                1, 1, 1, 3, 1, 3, 3, 0, 3, 1, 3, 0, 3, 3, 1, 1, 1, 3, 1, 3, 3, 3,
                1, 3, 1, 3, 3, 3, 0, 1, 1, 3, 1, 1, 3, 3, 1, 3, 1, 1, 0, 1, 0, 3,
                3, 1, 3, 3, 3, 3, 1, 1, 2, 3, 3, 1, 3, 1, 1, 1, 0, 3, 1, 3, 1, 3,
                2, 1, 3, 1, 3, 3, 2, 1, 4, 2, 3, 3, 3, 2, 1, 1, 4, 3, 3, 0, 3, 3,
                3, 1, 1, 3, 3, 3, 1, 3, 3, 1, 3, 3, 3, 1, 1, 1, 3, 3, 3, 1, 1, 1,
                3, 4])
```

```
data2.head()
```

	Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversityJobFairID	PayRate	Termd	Positic
0	Brown, Mia	1.103024e+09	1.0	1.0	0.0	1.0	1.0	3.0	1.0	28.50	0.0	
1	LaRotonda, William	1.106027e+09	0.0	2.0	1.0	1.0	1.0	3.0	0.0	23.00	0.0	
2	Steans, Tyrone	1.302053e+09	0.0	0.0	1.0	1.0	1.0	3.0	0.0	29.00	0.0	
3	Howard, Estelle	1.211051e+09	1.0	1.0	0.0	1.0	1.0	3.0	0.0	21.50	1.0	
4	Singh, Nan	1.307060e+09	0.0	0.0	0.0	1.0	1.0	3.0	0.0	16.56	0.0	

Масштабирование данных

```
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()
sc_data = sc.fit_transform(data2[['EngagementSurvey']])
plt.hist(data2['EngagementSurvey'], 10)
plt.show()
```




```
plt.hist(sc_data, 10)
plt.show()
```