Московский государственный технический университет им. Н.Э.

Баумана Кафедра «Системы обработки информации и управления»

Лабораторная работа №1

по дисциплине

«Методы машинного обучения»

на тему

«Обработка признаков(часть 1)»

Выполнил:

студент группы ИУ5-22М

Чжан Аньци

Москва — 2022 г.

1. Цель лабораторной работы:

изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

2. Задание

- устранение пропусков в данных;
- кодирование категориальных признаков;
- нормализацию числовых признаков.

3. Ход выполнения работы

Подключим необходимые библиотеки и настроим отображение графиков

In [9]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
sns.set(style='ticks')
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('retina')
pd.set_option('display.width',70)
```

Возьмём набор данных:

In [39]:
```
dataset=pd.read_csv('Life Expectancy Data.csv')
```

Посмотрим на эти наборы данных:

```
In [40]: dataset.dtypes
```

```
Out[40]: Country                          object
         Year                             int64
         Status                          object
         Life expectancy                float64
         Adult Mortality                float64
         infant deaths                    int64
         Alcohol                        float64
         percentage expenditure         float64
         Hepatitis B                    float64
         Measles                          int64
          BMI                           float64
         under-five deaths                int64
         Polio                          float64
         Total expenditure              float64
         Diphtheria                     float64
          HIV/AIDS                      float64
         GDP                            float64
         Population                     float64
          thinness  1-19 years          float64
          thinness 5-9 years            float64
         Income composition of resources float64
         Schooling                      float64
         dtype: object
```

```
In [41]: dataset.head()
```

Out[41]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GD
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1	584.2592
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1	612.6965
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1	631.7449
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1	669.9590
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1	63.5372

5 rows × 22 columns

```
In [42]: dataset.isnull().sum()
```

```
Out[42]: Country                          0
         Year                             0
         Status                           0
         Life expectancy                 10
         Adult Mortality                 10
         infant deaths                    0
         Alcohol                        194
         percentage expenditure           0
         Hepatitis B                    553
         Measles                          0
          BMI                            34
         under-five deaths                0
         Polio                           19
         Total expenditure              226
         Diphtheria                      19
          HIV/AIDS                        0
         GDP                            448
         Population                     652
          thinness  1-19 years           34
          thinness 5-9 years             34
         Income composition of resources 167
         Schooling                      163
         dtype: int64
```

3.1. Обработка пропусков в данных

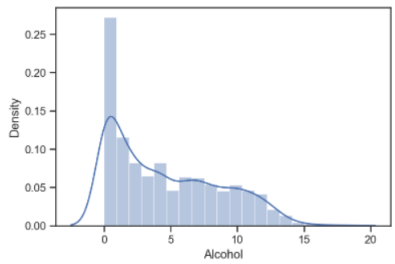
Будем работать с колонкой «Alcohol» и «Population».

Самый простой вариант — заполнить пропуски нулями:

In [43]:
```
from sklearn.impute import SimpleImputer

sns.distplot(dataset['Alcohol'])
```

Out[43]: <AxesSubplot:xlabel='Alcohol', ylabel='Density'>

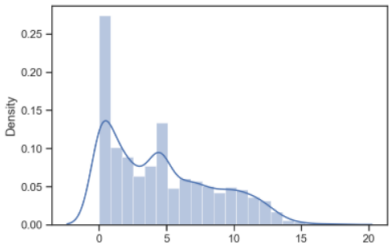


In [17]:
```
imp_mean=SimpleImputer(strategy='mean')
imp_freq=SimpleImputer(strategy='most_frequent')
imp_median=SimpleImputer(strategy='median')
```

Средний рейтинг:

In [47]:
```
alc_mean=imp_mean.fit_transform(dataset[['Alcohol']])
sns.distplot(seventyindicator_mean)
```

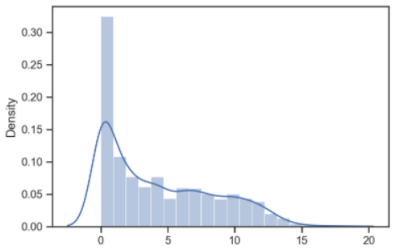
Out[47]: <AxesSubplot:ylabel='Density'>



Самый частый рейтинг:

In [48]:
```
alc_freq=imp_freq.fit_transform(dataset[['Alcohol']])
sns.distplot(seventyindicator_freq)
```

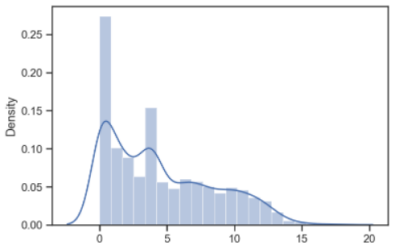
Out[48]: <AxesSubplot:ylabel='Density'>



Медианный рейтинг:

In [49]:
```
alc_median=imp_median.fit_transform(dataset[['Alcohol']])
sns.distplot(seventyindicator_median)
```

Out[49]: <AxesSubplot:ylabel='Density'>



Выбираем самый частый рейтинг:

In [51]:
```
dataset['Alcohol']=alc_freq
dataset['Population']=imp_freq.fit_transform(dataset[['Population']])
```

3.2. Кодирование категориальных признаков

Подключим библиотеку:

In [52]:
```
import sklearn.preprocessing
```

Рассмотрим колонку Country и Year

In [53]:
```
countrys=dataset['Country'].dropna().astype(str)
countrys.value_counts()
```

Out[53]:
```
Afghanistan              16
Peru                     16
Nicaragua                16
Niger                    16
Nigeria                  16
                         ..
Niue                      1
San Marino                1
Nauru                     1
Saint Kitts and Nevis     1
Dominica                  1
Name: Country, Length: 193, dtype: int64
```

In [54]:
```
years=dataset['Year'].dropna().astype(str)
years.value_counts()
```

Out[54]:
```
2013    193
2015    183
2014    183
2012    183
2011    183
2010    183
2009    183
2008    183
2007    183
2006    183
2005    183
2004    183
2003    183
2002    183
2001    183
2000    183
Name: Year, dtype: int64
```

Выполним кодирование категорий целочисленными значениями:

In [55]:
```
le=sklearn.preprocessing.LabelEncoder()
country_le=le.fit_transform(countrys)
print(np.unique(country_le))
le.inverse_transform(np.unique(country_le))
```

```
[  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107
 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
 180 181 182 183 184 185 186 187 188 189 190 191 192]
```

Out[55]:
```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola',
       'Antigua and Barbuda', 'Argentina', 'Armenia', 'Australia',
       'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain', 'Bangladesh',
       'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bhutan',
       'Bolivia (Plurinational State of)', 'Bosnia and Herzegovina',
       'Botswana', 'Brazil', 'Brunei Darussalam', 'Bulgaria',
       'Burkina Faso', 'Burundi', 'Cabo Verde', 'Cambodia', 'Cameroon',
       'Canada', 'Central African Republic', 'Chad', 'Chile', 'China',
       'Colombia', 'Comoros', 'Congo', 'Cook Islands', 'Costa Rica',
       'Croatia', 'Cuba', 'Cyprus', 'Czechia', "Côte d'Ivoire",
       "Democratic People's Republic of Korea",
       'Democratic Republic of the Congo', 'Denmark', 'Djibouti',
       'Dominica', 'Dominican Republic', 'Ecuador', 'Egypt',
```

```
'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',
'Ethiopia', 'Fiji', 'Finland', 'France', 'Gabon', 'Gambia',
'Georgia', 'Germany', 'Ghana', 'Greece', 'Grenada', 'Guatemala',
'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti', 'Honduras',
'Hungary', 'Iceland', 'India', 'Indonesia',
'Iran (Islamic Republic of)', 'Iraq', 'Ireland', 'Israel', 'Italy',
'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati',
'Kuwait', 'Kyrgyzstan', "Lao People's Democratic Republic",
'Latvia', 'Lebanon', 'Lesotho', 'Liberia', 'Libya', 'Lithuania',
'Luxembourg', 'Madagascar', 'Malawi', 'Malaysia', 'Maldives',
'Mali', 'Malta', 'Marshall Islands', 'Mauritania', 'Mauritius',
'Mexico', 'Micronesia (Federated States of)', 'Monaco', 'Mongolia',
'Montenegro', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia',
'Nauru', 'Nepal', 'Netherlands', 'New Zealand', 'Nicaragua',
'Niger', 'Nigeria', 'Niue', 'Norway', 'Oman', 'Pakistan', 'Palau',
'Panama', 'Papua New Guinea', 'Paraguay', 'Peru', 'Philippines',
'Poland', 'Portugal', 'Qatar', 'Republic of Korea',
'Republic of Moldova', 'Romania', 'Russian Federation', 'Rwanda',
'Saint Kitts and Nevis', 'Saint Lucia',
'Saint Vincent and the Grenadines', 'Samoa', 'San Marino',
'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia',
'Seychelles', 'Sierra Leone', 'Singapore', 'Slovakia', 'Slovenia',
'Solomon Islands', 'Somalia', 'South Africa', 'South Sudan',
'Spain', 'Sri Lanka', 'Sudan', 'Suriname', 'Swaziland', 'Sweden',
'Switzerland', 'Syrian Arab Republic', 'Tajikistan', 'Thailand',
'The former Yugoslav republic of Macedonia', 'Timor-Leste', 'Togo',
'Tonga', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
'Turkmenistan', 'Tuvalu', 'Uganda', 'Ukraine',
'United Arab Emirates',
'United Kingdom of Great Britain and Northern Ireland',
'United Republic of Tanzania', 'United States of America',
'Uruguay', 'Uzbekistan', 'Vanuatu',
'Venezuela (Bolivarian Republic of)', 'Viet Nam', 'Yemen',
'Zambia', 'Zimbabwe'], dtype=object)
```

In [56]:
```
le=sklearn.preprocessing.LabelEncoder()
years_le=le.fit_transform(years)
print(np.unique(years_le))
le.inverse_transform(np.unique(years_le))
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
```

Out[56]:
```
array(['2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007',
       '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015'],
      dtype=object)
```

Выполним кодирование категорий наборами бинарных значений:

In [57]:
```
country_oh=pd.get_dummies(countrys)
country_oh.head()
```

Out[57]:

	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	...	United Republic of Tanzania	United States of America	Uruguay	Uzbekistan	Vanuatu	(
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
1	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
3	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
4	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

5 rows × 193 columns

In [58]:
```
year_oh=pd.get_dummies(years)
year_oh.head()
```

Out[58]:

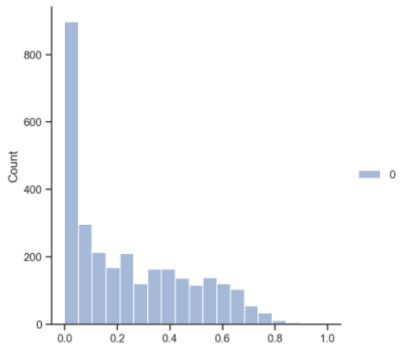
	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

3.3. Нормализацию числовых признаков.

MinMax-масштабирование:

In [61]:
```
minmax=sklearn.preprocessing.MinMaxScaler()
sns.displot(minmax.fit_transform(dataset[['Alcohol']]))
```

Out[61]: <seaborn.axisgrid.FacetGrid at 0x1a24ec44d30>



Масштабирование на основе Z-оценки:

In [62]:
```
stansc=sklearn.preprocessing.StandardScaler()
sns.displot(stansc.fit_transform(dataset[['Alcohol']]))
```

Out[62]: <seaborn.axisgrid.FacetGrid at 0x1a24c7dda00>