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

Лабораторная работа №2
по дисциплине

«Методы машинного обучения»

на тему

«Обработка признаков»

Выполнил:
студент группы ИУ5-24М
вань хао

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 [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [20]:
```
import seaborn as sns
%matplotlib inline
```

In [21]:
```
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 [4]:
```
dataset = pd.read_csv('GlobalLandTemperaturesByCountry.csv')
```

Посмотрим на эти наборы данных:

In [12]:
```
dataset.dtypes
```

Out[12]:
```
dt                              object
AverageTemperature             float64
AverageTemperatureUncertainty  float64
Country                         object
dtype: object
```

In [14]:
```
dataset.head()
```

Out[14]:

	dt	AverageTemperature	AverageTemperatureUncertainty	Country
0	1743-11-01	4.384	2.294	Åland
1	1743-12-01	NaN	NaN	Åland
2	1744-01-01	NaN	NaN	Åland
3	1744-02-01	NaN	NaN	Åland
4	1744-03-01	NaN	NaN	Åland

In [17]:
```
dataset.isnull().sum()
```

Out[17]:
```
dt                                 0
AverageTemperature             32651
AverageTemperatureUncertainty  31912
Country                            0
dtype: int64
```

3.1. Обработка пропусков в данных

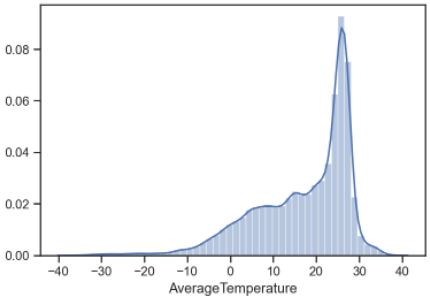
Будем работать с колонкой AverageTempareture и AverageTemperatureUncertainty.

Самый простой вариант — заполнить пропуски нулями:

In [7]:
```
from sklearn.impute import SimpleImputer
```

In [23]:
```
sns.distplot(dataset['AverageTemperature'])
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cde7024e50>
```

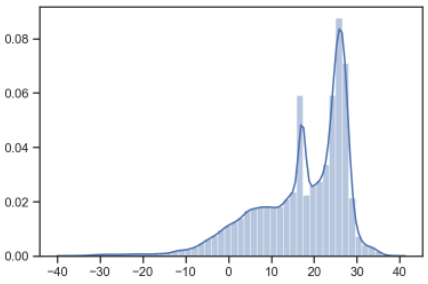


In [19]:
```
imp_mean=SimpleImputer(strategy='mean')
imp_freq=SimpleImputer(strategy='most_frequent')
imp_median=SimpleImputer(strategy='median')
```

Средний рейтинг:

In [25]:
```
avetemp_mean=imp_mean.fit_transform(dataset[['AverageTemperature']])
sns.distplot(avetemp_mean)
```

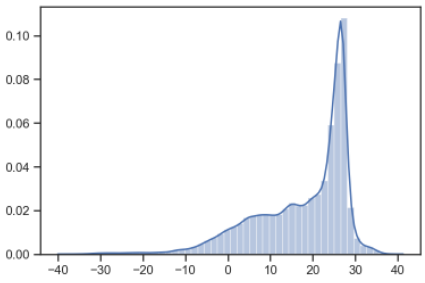
Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cde72faaf0>
```



Самый частый рейтинг:

In [26]:
```
avetemp_freq=imp_freq.fit_transform(dataset[['AverageTemperature']])
sns.distplot(avetemp_freq)
```

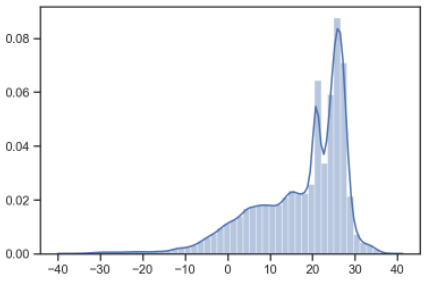
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x1cde72fa3d0>



Медианный рейтинг:

In [27]:
```
avetemp_median=imp_median.fit_transform(dataset[['AverageTemperature']])
sns.distplot(avetemp_median)
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x1cde763ce80>



Выбираем самый частый рейтинг:

In [29]:
```
dataset['AverageTemperature']=avetemp_freq
dataset['AverageTemperatureUncertainty']=imp_freq.fit_transform(dataset[['AverageTemperatureUncertainty']])
```

3.2. Кодирование категориальных признаков

Подключим библиотеку:

In [30]:
```
import sklearn.preprocessing
```

Рассмотрим колонку Country и dt:

In [32]:
```
countrys=dataset['Country'].dropna().astype(str)
countrys.value_counts()
```

Out[32]:
```
Europe                                3239
Faroe Islands                         3239
Macedonia                             3239
Spain                                 3239
United Kingdom                        3239
                                      ...
Guam                                  1329
Northern Mariana Islands              1329
French Southern And Antarctic Lands    788
Heard Island And Mcdonald Islands      788
Antarctica                             764
Name: Country, Length: 243, dtype: int64
```

In [33]:
```
years=dataset['dt'].dropna().astype(str)
years.value_counts()
```

Out[33]:
```
1964-06-01    243
1977-07-01    243
1961-02-01    243
2009-02-01    243
2009-12-01    243
              ...
1745-05-01     50
1745-08-01     50
1746-01-01     50
1746-12-01     50
1752-06-01     50
Name: dt, Length: 3239, dtype: int64
```

Выполним кодирование категорий целочисленными значениями:

In [36]:
```
le=sklearn.preprocessing.LabelEncoder()
countrys_le=le.fit_transform(countrys)
print(np.unique(countrys_le))
le.inverse_transform(np.unique(countrys_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
 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
 234 235 236 237 238 239 240 241 242]
```

Out[36]:
```
array(['Afghanistan', 'Africa', 'Albania', 'Algeria', 'American Samoa',
       'Andorra', 'Angola', 'Anguilla', 'Antarctica',
       'Antigua And Barbuda', 'Argentina', 'Armenia', 'Aruba', 'Asia',
       'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
       'Baker Island', 'Bangladesh', 'Barbados', 'Belarus', 'Belgium',
       'Belize', 'Benin', 'Bhutan', 'Bolivia',
       'Bonaire, Saint Eustatius And Saba', 'Bosnia And Herzegovina',
       'Botswana', 'Brazil', 'British Virgin Islands', 'Bulgaria',
       'Burkina Faso', 'Burma', 'Burundi', 'Cambodia', 'Cameroon',
       'Canada', 'Cape Verde', 'Cayman Islands',
       'Central African Republic', 'Chad', 'Chile', 'China',
       'Christmas Island', 'Colombia', 'Comoros', 'Congo',
       'Congo (Democratic Republic Of The)', 'Costa Rica', 'Croatia',
       'Cuba', 'Cura\x87ao', 'Cyprus', 'Czech Republic', "Côte D'Ivoire",
       'Denmark', 'Denmark (Europe)', 'Djibouti', 'Dominica',
       'Dominican Republic', 'Ecuador', 'Egypt', 'El Salvador',
       'Equatorial Guinea', 'Eritrea', 'Estonia', 'Ethiopia', 'Europe',
       'Falkland Islands (Islas Malvinas)', 'Faroe Islands',
       'Federated States Of Micronesia', 'Fiji', 'Finland', 'France',
       'France (Europe)', 'French Guiana', 'French Polynesia',
       'French Southern And Antarctic Lands', 'Gabon', 'Gambia',
       'Gaza Strip', 'Georgia', 'Germany', 'Ghana', 'Greece', 'Greenland',
       'Grenada', 'Guadeloupe', 'Guam', 'Guatemala', 'Guernsey', 'Guinea',
       'Guinea Bissau', 'Guyana', 'Haiti',
       'Heard Island And Mcdonald Islands', 'Honduras', 'Hong Kong',
       'Hungary', 'Iceland', 'India', 'Indonesia', 'Iran', 'Iraq',
       'Ireland', 'Isle Of Man', 'Israel', 'Italy', 'Jamaica', 'Japan',
       'Jersey', 'Jordan', 'Kazakhstan', 'Kenya', 'Kingman Reef',
       'Kiribati', 'Kuwait', 'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon',
       'Lesotho', 'Liberia', 'Libya', 'Liechtenstein', 'Lithuania',
       'Luxembourg', 'Macau', 'Macedonia', 'Madagascar', 'Malawi',
       'Malaysia', 'Mali', 'Malta', 'Martinique', 'Mauritania',
       'Mauritius', 'Mayotte', 'Mexico', 'Moldova', 'Monaco', 'Mongolia',
       'Montenegro', 'Montserrat', 'Morocco', 'Mozambique', 'Namibia',
       'Nepal', 'Netherlands', 'Netherlands (Europe)', 'New Caledonia',
       'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Niue',
       'North America', 'North Korea', 'Northern Mariana Islands',
       'Norway', 'Oceania', 'Oman', 'Pakistan', 'Palau', 'Palestina',
       'Palmyra Atoll', 'Panama', 'Papua New Guinea', 'Paraguay', 'Peru',
       'Philippines', 'Poland', 'Portugal', 'Puerto Rico', 'Qatar',
       'Reunion', 'Romania', 'Russia', 'Rwanda', 'Saint Barthélemy',
       'Saint Kitts And Nevis', 'Saint Lucia', 'Saint Martin',
       'Saint Pierre And Miquelon', 'Saint Vincent And The Grenadines',
       'Samoa', 'San Marino', 'Sao Tome And Principe', 'Saudi Arabia',
```

```
 'Senegal', 'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore',
 'Sint Maarten', 'Slovakia', 'Slovenia', 'Solomon Islands',
 'Somalia', 'South Africa', 'South America',
 'South Georgia And The South Sandwich Isla', 'South Korea',
 'Spain', 'Sri Lanka', 'Sudan', 'Suriname',
 'Svalbard And Jan Mayen', 'Swaziland', 'Sweden', 'Switzerland',
 'Syria', 'Taiwan', 'Tajikistan', 'Tanzania', 'Thailand',
 'Timor Leste', 'Togo', 'Tonga', 'Trinidad And Tobago', 'Tunisia',
 'Turkey', 'Turkmenistan', 'Turks And Caicas Islands', 'Uganda',
 'Ukraine', 'United Arab Emirates', 'United Kingdom',
 'United Kingdom (Europe)', 'United States', 'Uruguay',
 'Uzbekistan', 'Venezuela', 'Vietnam', 'Virgin Islands',
 'Western Sahara', 'Yemen', 'Zambia', 'Zimbabwe', 'Åland'],
dtype=object)
```

In [37]:
```
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 ... 3236 3237 3238]
```

Out[37]:
```
array(['1743-11-01', '1743-12-01', '1744-01-01', ..., '2013-07-01',
       '2013-08-01', '2013-09-01'], dtype=object)
```

Выполним кодирование категорий наборами бинарных значений:

In [39]:
```
country_oh=pd.get_dummies(countrys)
country_oh.head()
```

Out[39]:

	Afghanistan	Africa	Albania	Algeria	American Samoa	Andorra	Angola	Anguilla	Antarctica	Antigua And Barbuda	...	Uruguay	Uzbekistan	Venezuela	Vietnam	Virgin Islands	W
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

5 rows × 243 columns

In [40]:
```
year_oh=pd.get_dummies(years)
year_oh.head()
```

Out[40]:

	1743-11-01	1743-12-01	1744-01-01	1744-02-01	1744-03-01	1744-04-01	1744-05-01	1744-06-01	1744-07-01	1744-08-01	...	2012-12-01	2013-01-01	2013-02-01	2013-03-01	2013-04-01	2013-05-01	2013-06-01	2013-07-01	2013-08-01	2013-09-01
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

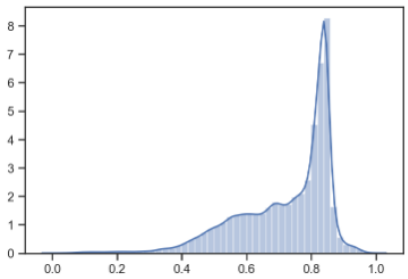
5 rows × 3239 columns

3.3. Нормализацию числовых признаков.

MinMax-масштабирование:

In [42]:
```
minmax=sklearn.preprocessing.MinMaxScaler()
sns.distplot(minmax.fit_transform(dataset[['AverageTemperature']]))
```

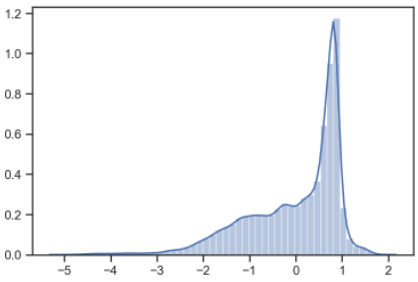
Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x1cde7706940>



Масштабирование на основе Z-оценки:

In [44]:
```
stansc=sklearn.preprocessing.StandardScaler()
sns.distplot(stansc.fit_transform(dataset[['AverageTemperature']]))
```

Out[44]: <matplotlib.axes._subplots.Axe...70>



Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Разведочный анализ данных. Исследование и визуализация данных» [Электронный ресурс]

[2] Wes McKinney «Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython»