Коваленко Артём ИУ5-64 Лабораторная № 3

Цель лабораторной работы

Изучение способов предварительной обработки данных для дальнейшего формирования моделей

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.

Ход выполнения лабораторной работы

Подключим необхоимые библиотеки и загрузим набор данных

In [1]:

```
import pandas as pd
import seaborn as sns
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder, MinMaxScaler, StandardScaler

%matplotlib inline

# Устанавливаем тип графиков
sns.set(style="ticks")

# Для лучшего качествоа графиков
from IPython.display import set_matplotlib_formats
set_matplotlib_formats("retina")

# Устанавливаем ширину экрана для отчета
pd.set_option("display.width", 70)

# Загружаем данные
data = pd.read_csv('googleplaystore.csv')
data.head()
```

Out[1]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone

In [2]:

```
data.shape
```

Out[2]:

```
(10841, 13)
```

1. Обработка пропусков в данных

In [3]:

```
# проверим есть ли пропущенные значения
data.isnull().sum()
```

Out[3]:

```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     1
Genres             0
Last Updated       0
Current Ver        8
Android Ver        3
dtype: int64
```

In [4]:

```
data.dtypes
```

Out[4]:

```
App                object
Category           object
Rating            float64
Reviews            object
Size               object
Installs           object
Type               object
Price              object
Content Rating     object
Genres             object
Last Updated       object
Current Ver        object
Android Ver        object
dtype: object
```

In [5]:

```
# Удаление колонок, содержащих пустые значения
data_new_1 = data.dropna(axis=1, how='any')
(data.shape, data_new_1.shape)
```

Out[5]:

```
((10841, 13), (10841, 8))
```

In [6]:

```
# Удаление строк, содержащих пустые значения
data_new_2 = data.dropna(axis=0, how='any')
(data.shape, data_new_2.shape)
```

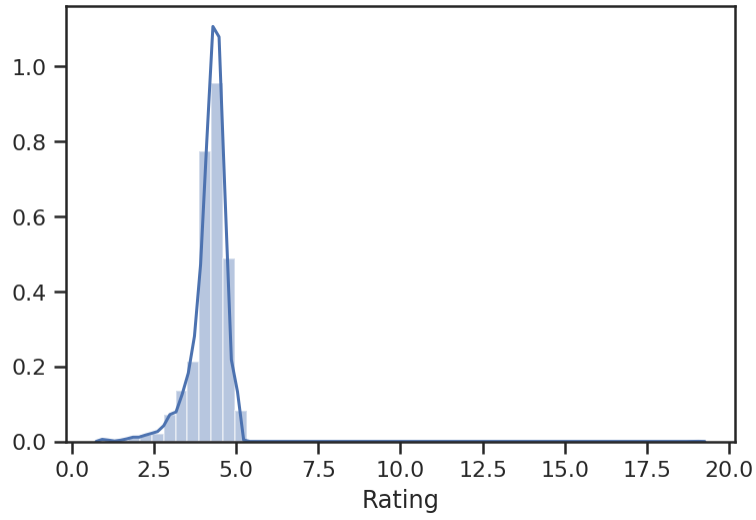
Out[6]:

```
((10841, 13), (9360, 13))
```

Будем работать с колонкой Rating

In [7]:

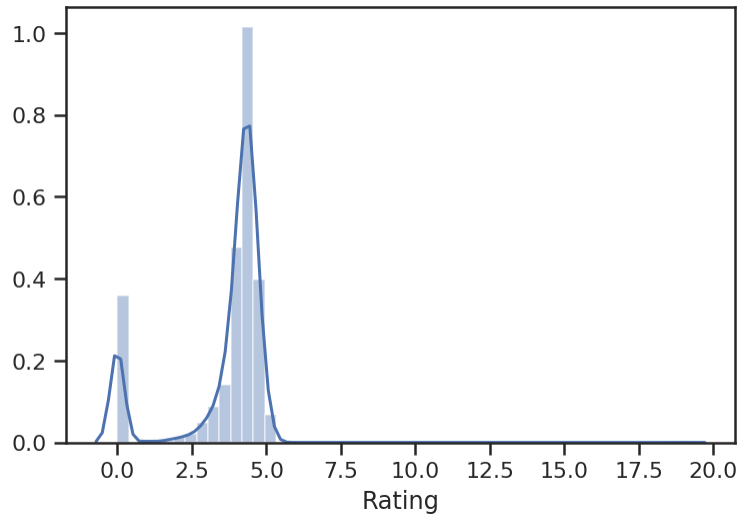
```
sns.distplot(data["Rating"]);
```



Самый простой способ - это заполнить нулями

In [8]:

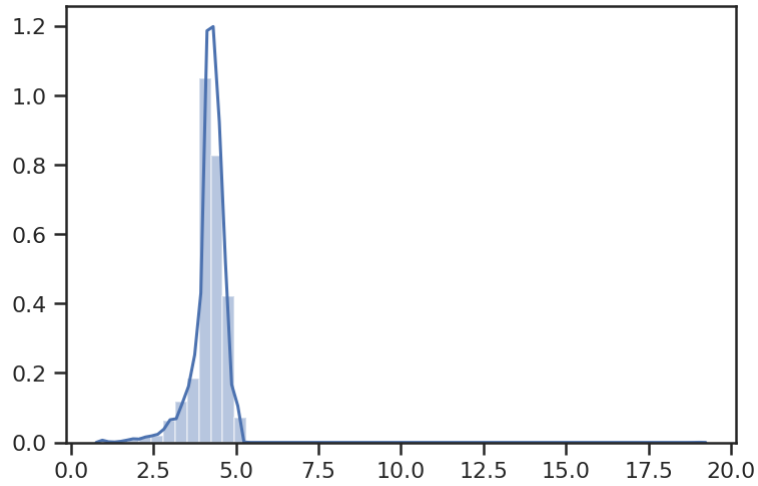
```
sns.distplot(data["Rating"].fillna(0));
```



Видно, что в данной ситуации это приводит к выбросам. Будем приложениям без рейтинга присваивать средний рейтинг

In [9]:

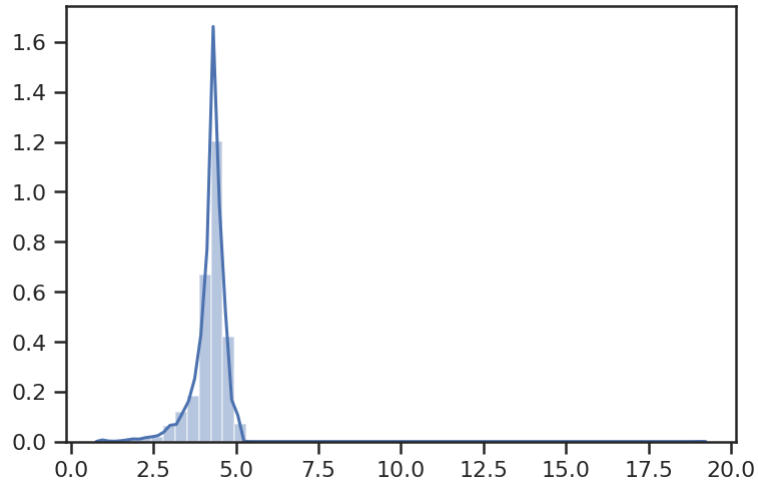
```
mean_imp = SimpleImputer(strategy="mean")
mean_rating = mean_imp.fit_transform(data[["Rating"]])
sns.distplot(mean_rating);
```



Попробуем заполнение медианой и самым частым значением

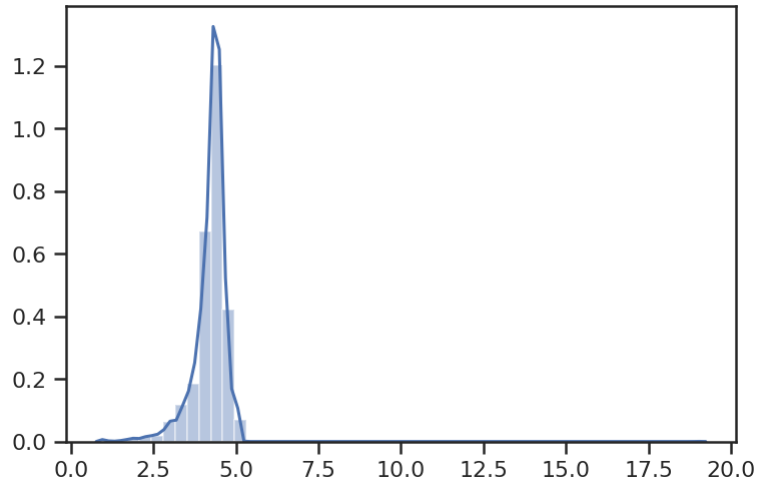
In [10]:

```
median_imp = SimpleImputer(strategy="median")
median_rating = median_imp.fit_transform(data[["Rating"]])
sns.distplot(median_rating);
```



In [11]:

```
most_freq_imp = SimpleImputer(strategy="most_frequent")
most_freq_rating = most_freq_imp.fit_transform(data[["Rating"]])
sns.distplot(most_freq_rating);
```



Будем использовать среднее значение

In [12]:

```
data["Rating"] = mean_rating
data["Rating"].isnull().sum()
```

Out[12]:

```
0
```

Как видим, у колонки Rating больше нет пропущенных значений

2. Кодирование категориальных признаков

Рассмотрим колонку Category

In [13]:

```
categories = data["Category"].dropna().astype(str)
categories.value_counts()
```

Out[13]:

```
FAMILY                 1972
GAME                   1144
TOOLS                   843
MEDICAL                 463
BUSINESS                460
PRODUCTIVITY            424
PERSONALIZATION         392
COMMUNICATION           387
SPORTS                  384
LIFESTYLE               382
FINANCE                 366
HEALTH_AND_FITNESS      341
PHOTOGRAPHY             335
SOCIAL                  295
NEWS_AND_MAGAZINES      283
SHOPPING                260
TRAVEL_AND_LOCAL        258
DATING                  234
BOOKS_AND_REFERENCE     231
VIDEO_PLAYERS           175
EDUCATION               156
ENTERTAINMENT           149
MAPS_AND_NAVIGATION     137
FOOD_AND_DRINK          127
HOUSE_AND_HOME           88
LIBRARIES_AND_DEMO       85
AUTO_AND_VEHICLES        85
WEATHER                  82
ART_AND_DESIGN           65
EVENTS                   64
PARENTING                60
COMICS                   60
BEAUTY                   53
1.9                       1
Name: Category, dtype: int64
```

In [14]:

```
le = LabelEncoder()
category_le = le.fit_transform(categories)
print(np.unique(category_le))
le.inverse_transform(np.unique(category_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]
```

Out[14]:

```
array(['1.9', 'ART_AND_DESIGN', 'AUTO_AND_VEHICLES', 'BEAUTY',
       'BOOKS_AND_REFERENCE', 'BUSINESS', 'COMICS', 'COMMUNICATION',
       'DATING', 'EDUCATION', 'ENTERTAINMENT', 'EVENTS', 'FAMILY',
       'FINANCE', 'FOOD_AND_DRINK', 'GAME', 'HEALTH_AND_FITNESS',
       'HOUSE_AND_HOME', 'LIBRARIES_AND_DEMO', 'LIFESTYLE',
       'MAPS_AND_NAVIGATION', 'MEDICAL', 'NEWS_AND_MAGAZINES',
       'PARENTING', 'PERSONALIZATION', 'PHOTOGRAPHY', 'PRODUCTIVITY',
       'SHOPPING', 'SOCIAL', 'SPORTS', 'TOOLS', 'TRAVEL_AND_LOCAL',
       'VIDEO_PLAYERS', 'WEATHER'], dtype=object)
```

In [15]:

```
data.head()
```

Out[15]:

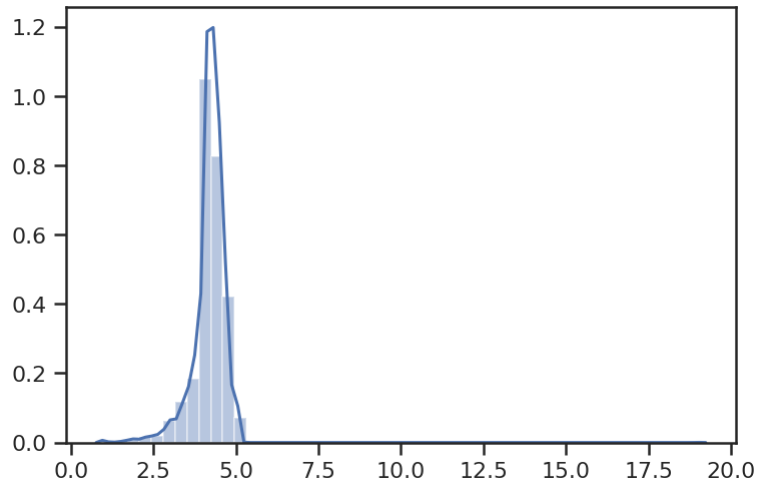
	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone

3. Масштабирование данных

Min-Max масштабирование

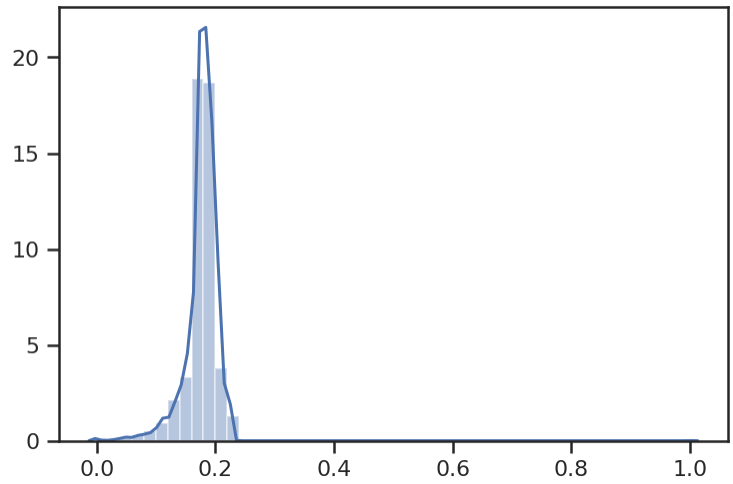
In [16]:

```
sns.distplot(data[["Rating"]]);
```



In [17]:

```
mm = MinMaxScaler()
sns.distplot(mm.fit_transform(data[["Rating"]]));
```



На основе Z-оценки

In [18]:

```
ss = StandardScaler()
sns.distplot(ss.fit_transform(data[["Rating"]]));
```