Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»

Отчет

по лабораторной работе № 3

по курсу «Технологии машинного обучения»

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

ИСПОЛНИТЕЛЬ:

Тарасов Владислав

Группа ИУ5-64Б

_____

"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

_____

"__"_____2020 г.

Москва 2020

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.

Дата-сет

In [17]:

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

data = pd.read_csv("../data/games.csv")
data.head()
```

Out[17]:

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22

Посмотрим на типы колонок

In [4]:

```
data.dtypes
```

Out[4]:

```
Name                object
Platform            object
Year_of_Release    float64
Genre               object
Publisher           object
NA_Sales           float64
EU_Sales           float64
JP_Sales           float64
Other_Sales        float64
Global_Sales       float64
Critic_Score       float64
Critic_Count       float64
User_Score          object
User_Count         float64
Developer           object
Rating              object
dtype: object
```

In [16]:

```
data.shape
```

Out[16]:

```
(16719, 16)
```

Обработка пропусков в данных

In [17]:

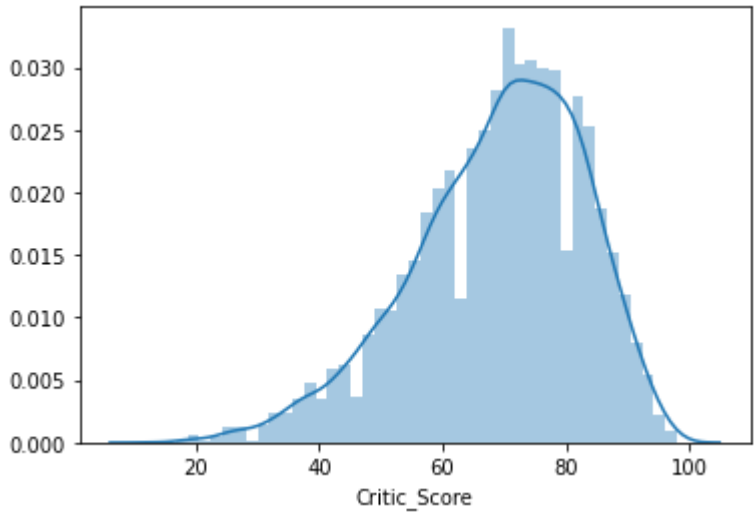
```
data.isnull().sum()
```

Out[17]:

```
Name                  2
Platform              0
Year_of_Release     269
Genre                 2
Publisher            54
NA_Sales              0
EU_Sales              0
JP_Sales              0
Other_Sales           0
Global_Sales          0
Critic_Score       8582
Critic_Count       8582
User_Score         6704
User_Count         9129
Developer          6623
Rating             6769
dtype: int64
```

In [21]:

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



In [52]:

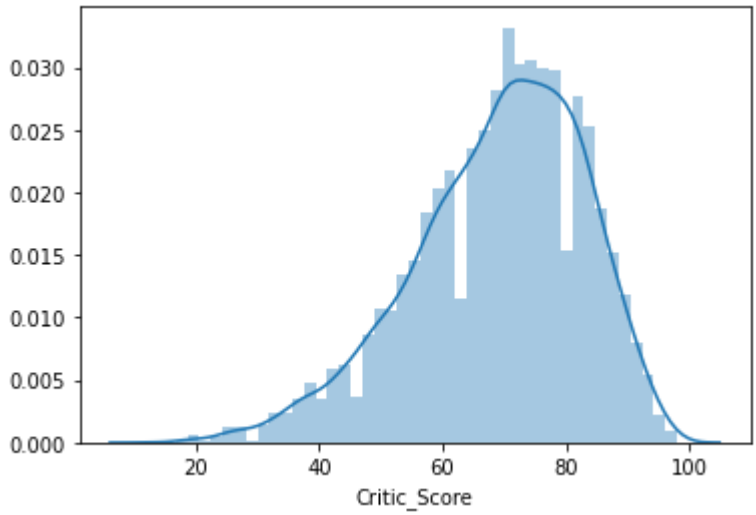
```
sns.distplot(data["Critic_Score"].fillna(0));
```

Out[52]:

```
0         76.0
1          0.0
2         82.0
3         80.0
4          0.0
          ...
16714      0.0
16715      0.0
16716      0.0
16717      0.0
16718      0.0
Name: Critic_Score, Length: 16719, dtype: float64
```

In [59]:

```
sns.distplot(data["Critic_Score"].dropna());
```

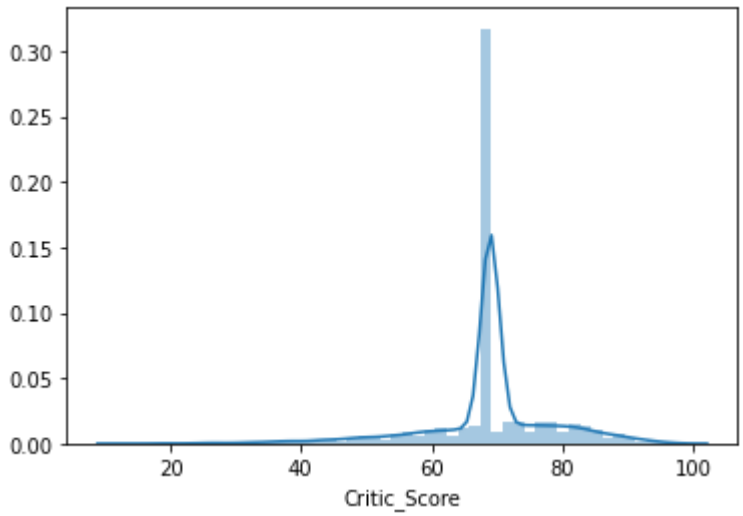


In [56]:

```
sns.distplot(data["Critic_Score"].fillna(data["Critic_Score"].mean()))
```

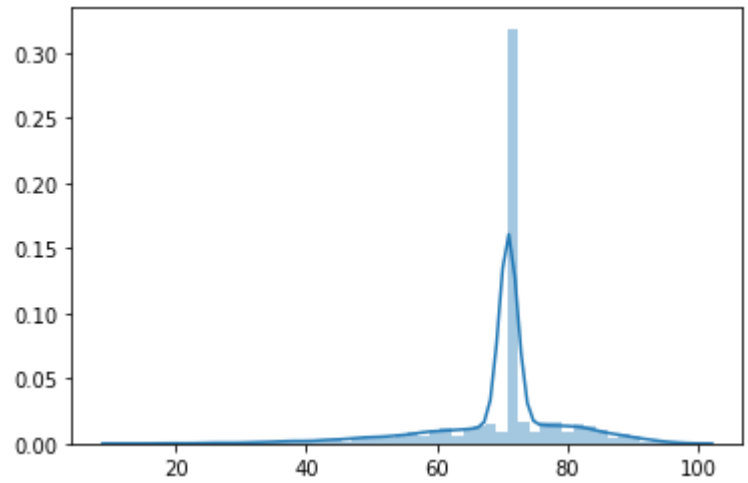
Out[56]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11fbb8c90>
```



In [12]:

```
mediana = sklearn.impute.SimpleImputer(strategy="median")
median_rating = mediana.fit_transform(data[["Critic_Score"]])
sns.distplot(median_rating);
```

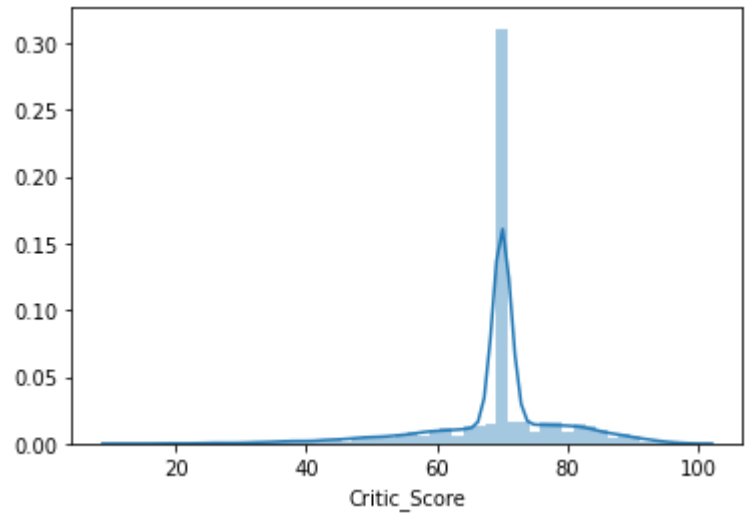


In [5]:

```
sns.distplot(data["Critic_Score"].fillna(70.0))
```

Out[5]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x121cd9e50>
```



In [13]:

```
data["Critic_Score"] = median_rating

data["Critic_Score"].isnull().sum()
```

Out[13]:

```
0
```

Как видим, у колонки Rating больше нет пропущенных значений

Кодирование категориальных признаков

In [15]:

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

Out[15]:

```
Action          3370
Sports          2348
Misc            1750
Role-Playing    1500
Shooter         1323
Adventure       1303
Racing          1249
Platform         888
Simulation       874
Fighting         849
Strategy         683
Puzzle           580
Name: Genre, dtype: int64
```

In [18]:

```
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]
```

Out[18]:

```
array(['Action', 'Adventure', 'Fighting', 'Misc', 'Platform', 'Puzzl
e',
       'Racing', 'Role-Playing', 'Shooter', 'Simulation', 'Sports',
       'Strategy'], dtype=object)
```

In [19]:

```
data.head()
```

Out[19]:

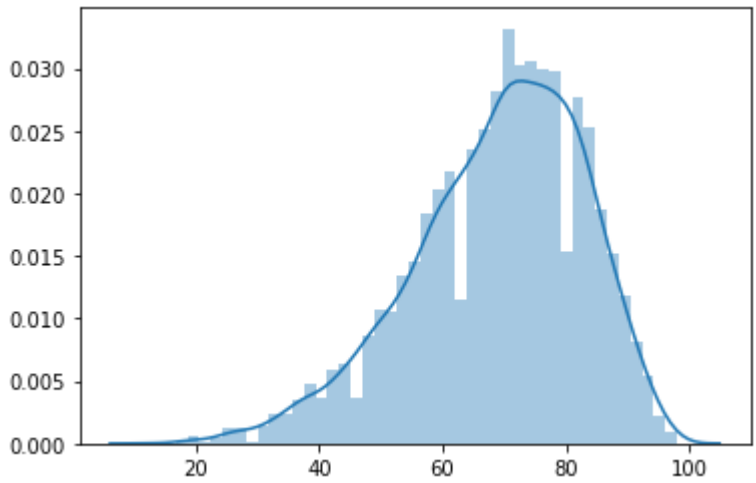
	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22

Масштабирование данных

min-max масштабирование

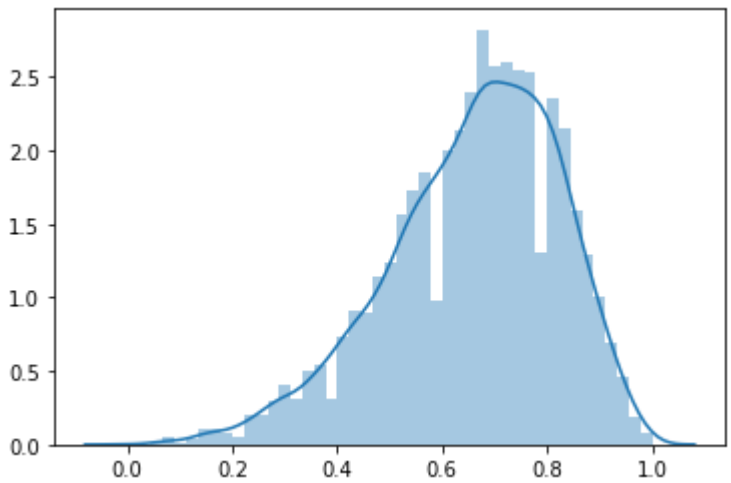
In [20]:

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



In [21]:

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



На основе Z-оценки

In [22]:

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