Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

Лабораторная работа №3

по дисциплине

«Методы машинного обучения»

на тему

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

Выполнил: студент группы

ИУ5-21М

Аунг Каунг Кхант

Москва — 2020 г.

1. Цель лабораторной работы
   Изучить способы предварительной обработки данных для дальнейшего формирования моделей [1

In [0]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import sklearn.impute
import sklearn.preprocessing
%matplotlib inline
sns.set(style="ticks")
from IPython.display import set_matplotlib_formats
set_matplotlib_formats("retina")
```

In [0]:
```
pd.set_option("display.width", 70)
```

In [0]:
```
data = pd.read_csv("googleplaystore.csv")
```

In [9]:
```
data.head()
```

Out[9]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Cu Ve
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0

In [10]:
```
data.dtypes
```

Out[10]:
```
App               object
Category          object
Rating           float64
Reviews            int64
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 [11]:
```
data.shape
```

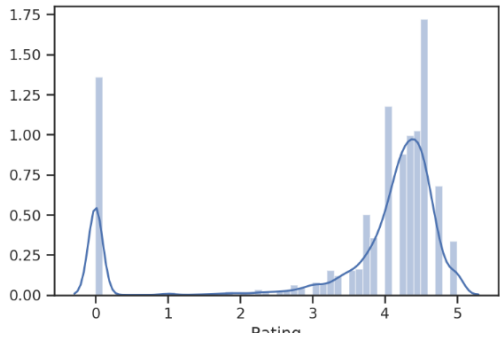
Out[11]:
```
(10841, 13)
```

]

In [12]:
```
data.isnull().sum()
```

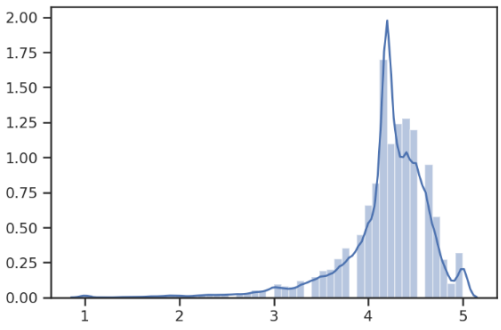
Out[12]:
```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     0
Genres             1
Last Updated       0
Current Ver        8
Android Ver        2
dtype: int64
```

In [13]:
```
sns.distplot(data["Rating"].fillna(0));
```

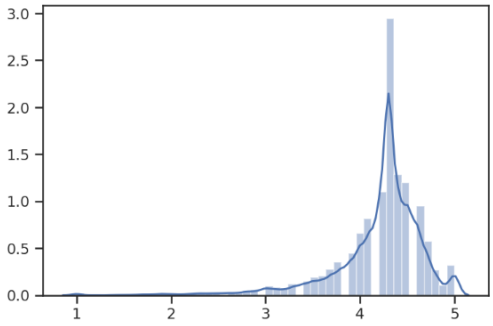


-

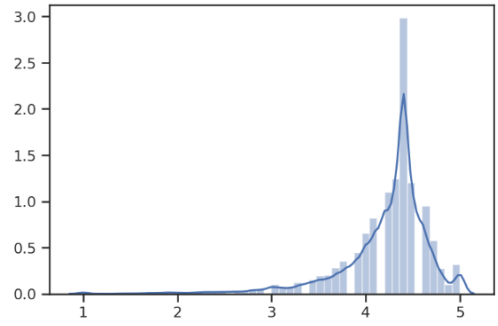
In [14]:
```
mean_imp = sklearn.impute.SimpleImputer(strategy="mean")
mean_rat = mean_imp.fit_transform(data[["Rating"]])
sns.distplot(mean_rat);
```



In [15]:
```
med_imp = sklearn.impute.SimpleImputer(strategy="median")
med_rat = med_imp.fit_transform(data[["Rating"]])
sns.distplot(med_rat);
```



In [16]:
```
freq_imp = sklearn.impute.SimpleImputer(strategy="most_frequent")
freq_rat = freq_imp.fit_transform(data[["Rating"]])
sns.distplot(freq_rat);
```



In [20]:
```
types = data["Type"].dropna().astype(str)
types.value_counts()
```

Out[20]:
```
Free    10040
Paid      800
Name: Type, dtype: int64
```

In [21]:
```
le = sklearn.preprocessing.LabelEncoder()
type_le = le.fit_transform(types)
print(np.unique(type_le))
le.inverse_transform(np.unique(type_le))
```

```
[0 1]
```

Out[21]:
```
array(['Free', 'Paid'], dtype=object)
```

In [23]:
```
type_oh = pd.get_dummies(types)
type_oh.head()
```

Out[23]:

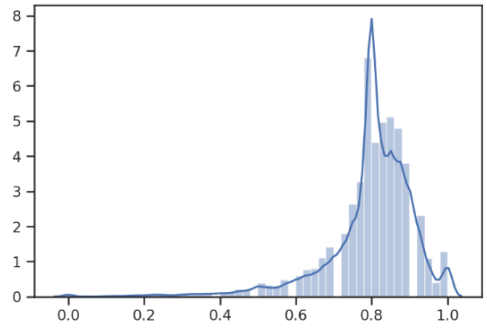
	Free	Paid
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0

In [24]:
```
type_oh[type_oh["Paid"] == 1].head()
```

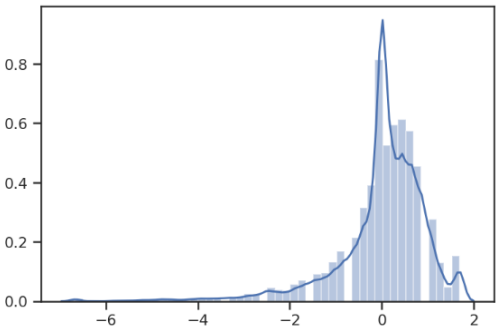
Out[24]:

	Free	Paid
234	0	1
235	0	1
290	0	1
291	0	1
427	0	1

In [25]:
```
mm = sklearn.preprocessing.MinMaxScaler()
sns.distplot(mm.fit_transform(data[["Rating"]]));
```



In [26]:
```
ss = sklearn.preprocessing.StandardScaler()
sns.distplot(ss.fit_transform(data[["Rating"]]));
```



Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Обработка пропусков в данных, кодиро- вание категориальных признаков, масштабирование данных» [Электронный ре- сурс] // GitHub. — 2019. — Режим доступа: https://github.com/ugapanyuk/ml_ course/wiki/LAB_MISSING (дата обращения: 05.04.2019).

[2] Team The IPython Development. IPython 7.3.0 Documentation [Electronic resource] // Read the Docs. — 2019. — Access mode: https://ipython.readthedocs.io/en/ stable/ (online; accessed: 20.02.2019).

[3] Waskom M. seaborn 0.9.0 documentation [Electronic resource] // PyData. — 2018. — Access mode: https://seaborn.pydata.org/ (online; accessed: 20.02.2019).

[4] pandas 0.24.1 documentation [Electronic resource] // PyData. — 2019. — Access mode: http://pandas.pydata.org/pandas-docs/stable/ (online; accessed: 20.02.2019).

[5] Gupta L. Google Play Store Apps [Electronic resource] // Kaggle. — 2019. — Access mode: https://www.kaggle.com/lava18/google-play-store-apps (online; accessed: 05.04.2019).