TMO-RK1

April 18, 2021

0.0.1 №1

0.0.2 , 5-62 , 19. 3.

№3. ( )
(label encoding, one hot encoding) .
? #### 5-62

[1]:
```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import *
```

0.1

[2]:
```python
data = pd.read_csv('marvel-wikia-data.csv', sep=",")
```

[3]:
```python
#
data.head()
```

[3]:
```
   page_id                                 name  \
0     1678            Spider-Man (Peter Parker)   
1     7139      Captain America (Steven Rogers)   
2    64786  Wolverine (James \"Logan\" Howlett)   
3     1868    Iron Man (Anthony \"Tony\" Stark)   
4     2460                  Thor (Thor Odinson)   

                                   urlslug                ID  \
0              \/Spider-Man_(Peter_Parker)   Secret Identity   
1        \/Captain_America_(Steven_Rogers)   Public Identity   
2  \/Wolverine_(James_%22Logan%22_Howlett)   Public Identity   
3    \/Iron_Man_(Anthony_%22Tony%22_Stark)   Public Identity   
4                    \/Thor_(Thor_Odinson)  No Dual Identity   

                ALIGN         EYE         HAIR              SEX  GSM  \
0     Good Characters  Hazel Eyes  Brown Hair  Male Characters  NaN
1     Good Characters   Blue Eyes  White Hair  Male Characters  NaN
2  Neutral Characters   Blue Eyes  Black Hair  Male Characters  NaN
3     Good Characters   Blue Eyes  Black Hair  Male Characters  NaN
4     Good Characters   Blue Eyes  Blond Hair  Male Characters  NaN

               ALIVE  APPEARANCES FIRST APPEARANCE    Year
0  Living Characters       4043.0           Aug-62  1962.0
1  Living Characters       3360.0           Mar-41  1941.0
2  Living Characters       3061.0           Oct-74  1974.0
3  Living Characters       2961.0           Mar-63  1963.0
4  Living Characters       2258.0           Nov-50  1950.0
```

[4]:

```
#
data.dtypes
```

[4]:

```
page_id               int64
name                 object
urlslug              object
ID                   object
ALIGN                object
EYE                  object
HAIR                 object
SEX                  object
GSM                  object
ALIVE                object
APPEARANCES         float64
FIRST APPEARANCE     object
Year                float64
dtype: object
```

0.2

[5]:

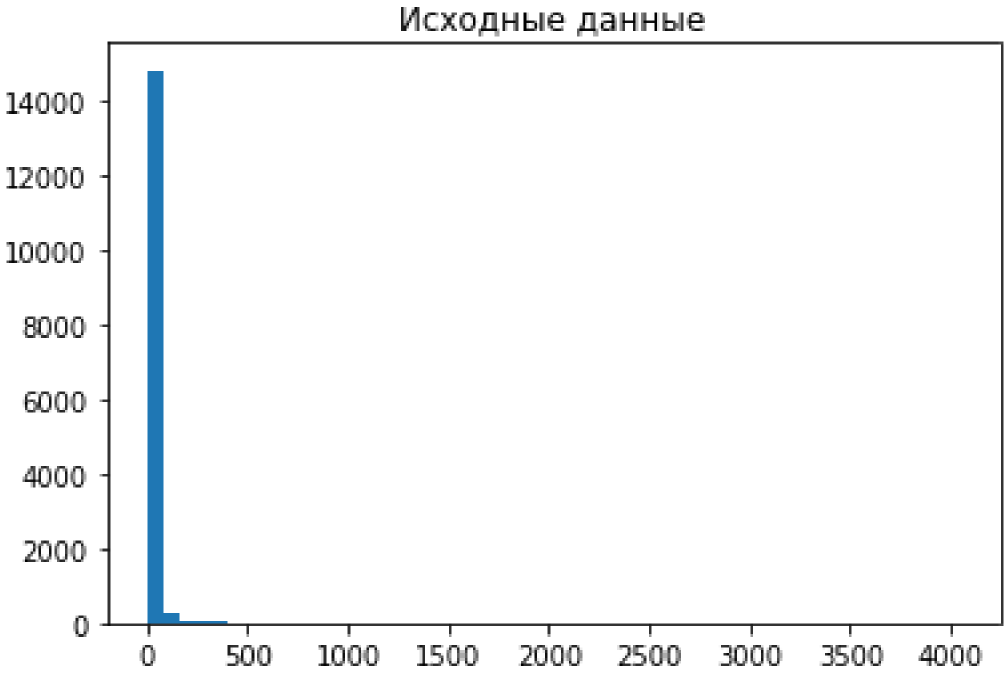
```
#
data.describe()
```

[5]:

```
             page_id   APPEARANCES          Year
count   16376.000000  15280.000000  15561.000000
mean   300232.082377     17.033377   1984.951803
std    253460.403399     96.372959     19.663571
min      1025.000000      1.000000   1939.000000
25%     28309.500000      1.000000   1974.000000
50%    282578.000000      3.000000   1990.000000
75%    509077.000000      8.000000   2000.000000
max    755278.000000   4043.000000   2013.000000
```

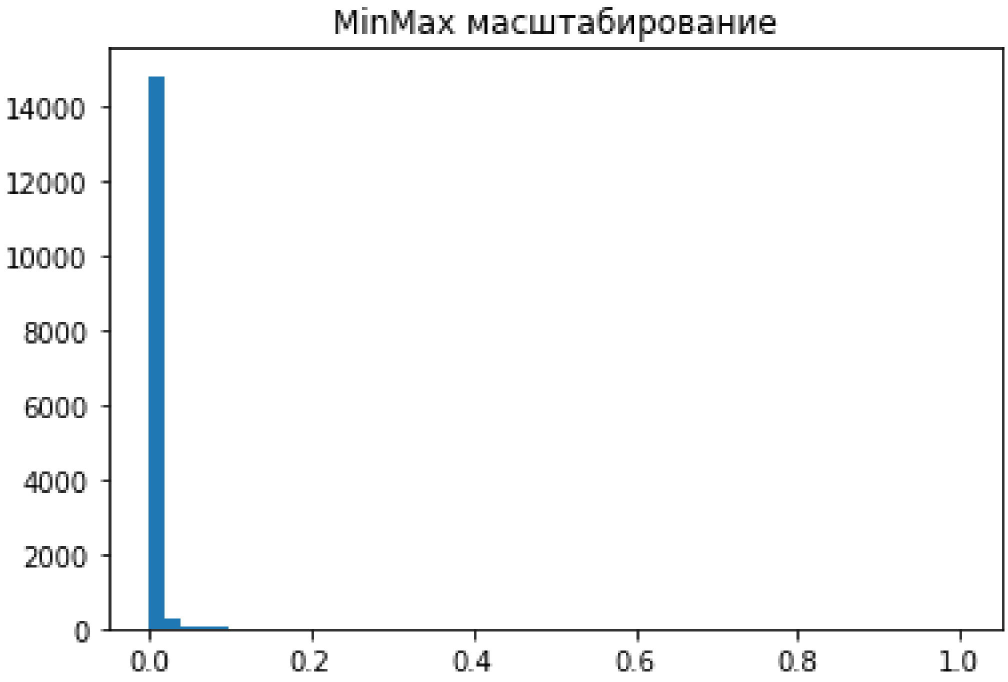
0.2.1 - 1 4043

[6]:
```
plt.hist(data['APPEARANCES'], 50)
plt.title("         ")
plt.show()
```



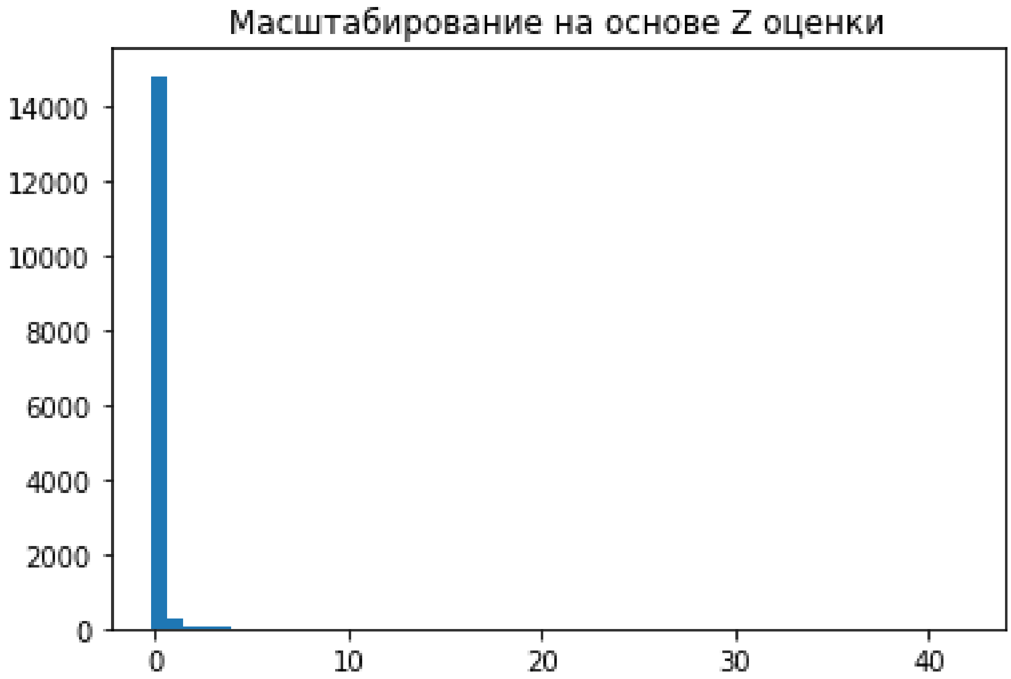
0.2.2 MinMax - 0 1

[7]:
```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['APPEARANCES']])
plt.hist(sc1_data, 50)
plt.title("MinMax         ")
plt.show()
```



0.2.3 Z - -3 3

[8]:
```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['APPEARANCES']])
plt.hist(sc2_data, 50)
plt.title("                Z    ")
plt.show()
```



0.3

0.3.1 Label encoding

[9]:
```
#                      "Unknown"
imp2 = SimpleImputer(missing_values=np.nan, strategy='constant',␣
 ↪fill_value='Unknown')
data['EYE'] = imp2.fit_transform(data[['EYE']])

#
types = data['EYE']
types.unique()
```

[9]:
```
array(['Hazel Eyes', 'Blue Eyes', 'Brown Eyes', 'Green Eyes', 'Grey Eyes',
       'Yellow Eyes', 'Gold Eyes', 'Red Eyes', 'Black Eyeballs',
       'Amber Eyes', 'Variable Eyes', 'Unknown', 'Black Eyes',
       'White Eyes', 'Orange Eyes', 'Silver Eyes', 'Purple Eyes',
       'Pink Eyes', 'One Eye', 'Violet Eyes', 'Multiple Eyes',
       'Magenta Eyes', 'Yellow Eyeballs', 'No Eyes', 'Compound Eyes'],
      dtype=object)
```

[10]:
```
#label encoding
le = LabelEncoder()
```

```
data_le = le.fit_transform(types)
```

```
[11]: np.unique(data_le)
```

```
[11]: array([ 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])
```

```
[12]: le.inverse_transform(data_le)
```

```
[12]: array(['Hazel Eyes', 'Blue Eyes', 'Blue Eyes', …, 'Black Eyes',
             'Unknown', 'Unknown'], dtype=object)
```

0.3.2 One hot encoding

```
[13]: pd.get_dummies(data['EYE']).head()
```

```
[13]:    Amber Eyes  Black Eyeballs  Black Eyes  Blue Eyes  Brown Eyes  \
      0           0               0           0          0           0
      1           0               0           0          1           0
      2           0               0           0          1           0
      3           0               0           0          1           0
      4           0               0           0          1           0

         Compound Eyes  Gold Eyes  Green Eyes  Grey Eyes  Hazel Eyes  …  \
      0              0          0           0          0           1  …
      1              0          0           0          0           0  …
      2              0          0           0          0           0  …
      3              0          0           0          0           0  …
      4              0          0           0          0           0  …

         Pink Eyes  Purple Eyes  Red Eyes  Silver Eyes  Unknown  Variable Eyes  \
      0          0            0         0            0        0              0
      1          0            0         0            0        0              0
      2          0            0         0            0        0              0
      3          0            0         0            0        0              0
      4          0            0         0            0        0              0

         Violet Eyes  White Eyes  Yellow Eyeballs  Yellow Eyes
      0            0           0                0            0
      1            0           0                0            0
      2            0           0                0            0
      3            0           0                0            0
      4            0           0                0            0
```

```
[5 rows x 25 columns]
```

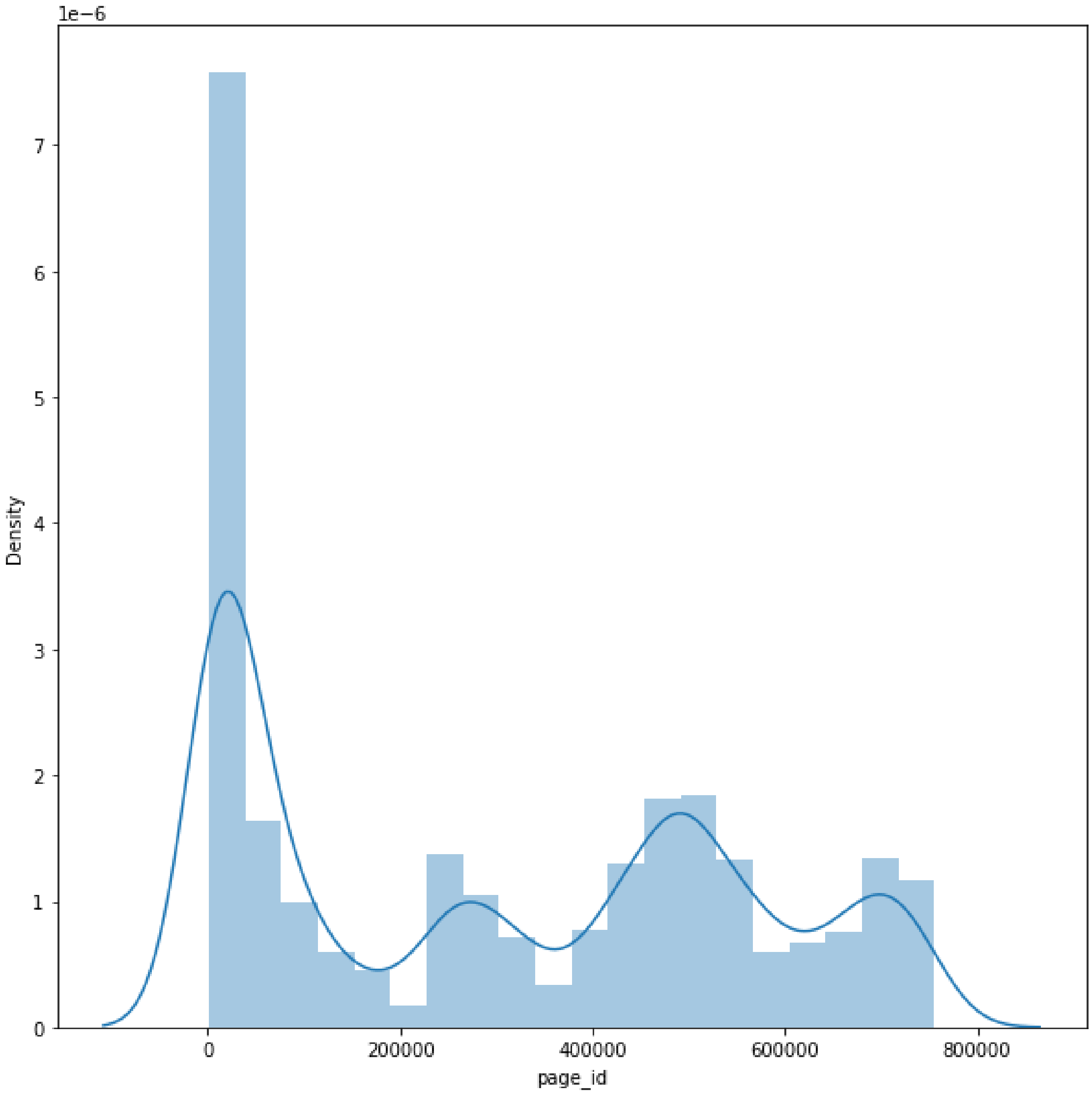
0.4

, .

```
[14]: fig, ax = plt.subplots(figsize=(10,10))
      sns.distplot(data['page_id'])
```

```
/usr/local/lib/python3.9/site-packages/seaborn/distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

```
[14]: <AxesSubplot:xlabel='page_id', ylabel='Density'>
```


1e−6
7
6
5
4
3
2
1
0
Density
0
200000
400000
600000
800000
page_id

[ ]: