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

Лабораторная работа №3 по курсу «Методы машинного обучения»

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

Выполнил: Ильин В.С.
Группа ИУ5-22М

Москва – 2019

1. Описание задания

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.

2. Ход выполнения лабораторной работы

In [0]:
```
!pip install -U -q PyDrive
import os
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
```

In [0]:
```
# 1. Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

In [0]:
```
# choose a local (colab) directory to store the data.
local_download_path = os.path.expanduser('~/data')
try:
  os.makedirs(local_download_path)
except: pass
```

In [0]:
```
# 2. Auto-iterate using the query syntax
#    https://developers.google.com/drive/v2/web/search-parameters
file_list = drive.ListFile(
    {'q': "title='dc-wikia-data.csv'"}).GetList()
```

In [0]:
```
for f in file_list:
  # 3. Create & download by id.
  print('title: %s, id: %s' % (f['title'], f['id']))
  fname = os.path.join(local_download_path, f['title'])
  print('downloading to {}'.format(fname))
  f_ = drive.CreateFile({'id': f['id']})
  f_.GetContentFile(fname)
```

title: dc-wikia-data.csv, id: 1mp_Y-6OLZLTtpzYI8UA7-_EulThJZK-o
downloading to /root/data/dc-wikia-data.csv

In [0]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [0]:
```
data = pd.read_csv(fname, sep=",")
```

In [0]:
```
data.head()
```

Out[0]:
```
   page_id                          name                                 urlslug  \
0     1422           Batman (Bruce Wayne)          \/wiki\/Batman_(Bruce_Wayne)
1    23387          Superman (Clark Kent)         \/wiki\/Superman_(Clark_Kent)
2     1458     Green Lantern (Hal Jordan)    \/wiki\/Green_Lantern_(Hal_Jordan)
3     1659      James Gordon (New Earth)      \/wiki\/James_Gordon_(New_Earth)
4     1576   Richard Grayson (New Earth)   \/wiki\/Richard_Grayson_(New_Earth)

                ID             ALIGN         EYE         HAIR              SEX  \
0  Secret Identity   Good Characters   Blue Eyes   Black Hair  Male Characters
1  Secret Identity   Good Characters   Blue Eyes   Black Hair  Male Characters
2  Secret Identity   Good Characters  Brown Eyes   Brown Hair  Male Characters
3  Public Identity   Good Characters  Brown Eyes   White Hair  Male Characters
4  Secret Identity   Good Characters   Blue Eyes   Black Hair  Male Characters

   GSM              ALIVE  APPEARANCES FIRST APPEARANCE    YEAR
0  NaN  Living Characters       3093.0        1939, May  1939.0
1  NaN  Living Characters       2496.0    1986, October  1986.0
2  NaN  Living Characters       1565.0    1959, October  1959.0
3  NaN  Living Characters       1316.0   1987, February  1987.0
4  NaN  Living Characters       1237.0      1940, April  1940.0
```

In [0]:
```
data.shape
```

Out[0]:
```
(6896, 13)
```

In [0]:
```
data.dtypes
```

Out[0]:
```
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
```

In [0]: data.isnull().sum()

```
Out[0]: page_id               0
        name                  0
        urlslug               0
        ID                 2013
        ALIGN               601
        EYE                3628
        HAIR               2274
        SEX                 125
        GSM                6832
        ALIVE                 3
        APPEARANCES         355
        FIRST APPEARANCE     69
        YEAR                 69
        dtype: int64
```

2.1. Удаление

```
In [0]: # Удаление колонок
        data_new_1 = data.dropna(axis=1, how='any')
        (data.shape, data_new_1.shape)
```

Out[0]: ((6896, 13), (6896, 3))

```
In [0]: # Удаление строк
        data_new_2 = data.dropna(axis=0, how='any')
        (data.shape, data_new_2.shape)
```

Out[0]: ((6896, 13), (38, 13))

2.2. Заполнение нулями

```
In [0]: # Заполнение всех пропущенных значений нулями
        data_new_3 = data.fillna(0)
        data_new_3.head()
```

```
Out[0]:   page_id                          name                              urlslug  \
        0    1422         Batman (Bruce Wayne)         \/wiki\/Batman_(Bruce_Wayne)
        1   23387        Superman (Clark Kent)        \/wiki\/Superman_(Clark_Kent)
        2    1458   Green Lantern (Hal Jordan)   \/wiki\/Green_Lantern_(Hal_Jordan)
        3    1659    James Gordon (New Earth)     \/wiki\/James_Gordon_(New_Earth)
        4    1576  Richard Grayson (New Earth)  \/wiki\/Richard_Grayson_(New_Earth)

                       ID            ALIGN        EYE        HAIR              SEX  \
        0  Secret Identity  Good Characters  Blue Eyes  Black Hair  Male Characters
```

1	Secret Identity	Good Characters	Blue Eyes	Black Hair	Male Characters
2	Secret Identity	Good Characters	Brown Eyes	Brown Hair	Male Characters
3	Public Identity	Good Characters	Brown Eyes	White Hair	Male Characters
4	Secret Identity	Good Characters	Blue Eyes	Black Hair	Male Characters

	GSM	ALIVE	APPEARANCES	FIRST APPEARANCE	YEAR
0	0	Living Characters	3093.0	1939, May	1939.0
1	0	Living Characters	2496.0	1986, October	1986.0
2	0	Living Characters	1565.0	1959, October	1959.0
3	0	Living Characters	1316.0	1987, February	1987.0
4	0	Living Characters	1237.0	1940, April	1940.0

2.3. Внедрение значений (числовые данные)

In [0]:
```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
total_count = data.shape[0]
num_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'
              .format(col, dt, temp_null_count, temp_perc))
```

Колонка APPEARANCES. Тип данных float64. Количество пустых значений 355, 5.15%.
Колонка YEAR. Тип данных float64. Количество пустых значений 69, 1.0%.

In [0]:
```
data_num = data[num_cols]
data_num
```

Out[0]:

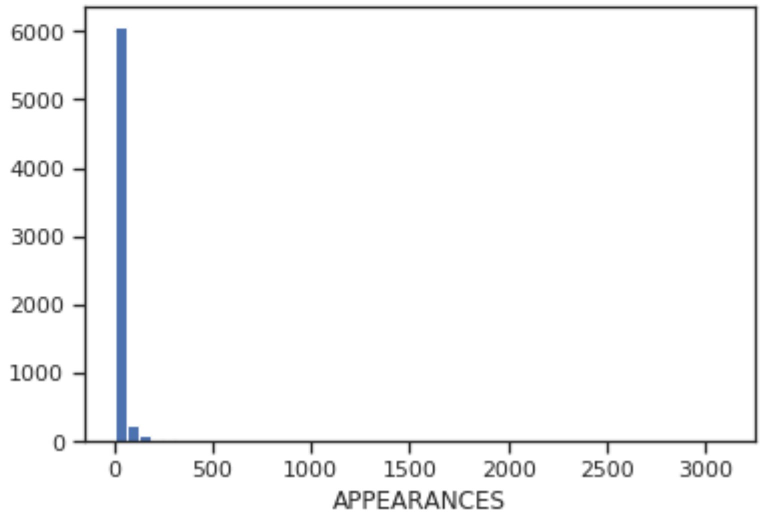
	APPEARANCES	YEAR
0	3093.0	1939.0
1	2496.0	1986.0
2	1565.0	1959.0
3	1316.0	1987.0
4	1237.0	1940.0
5	1231.0	1941.0
6	1121.0	1941.0
7	1095.0	1989.0
8	1075.0	1969.0
9	1028.0	1956.0
10	1028.0	1956.0
11	969.0	1940.0
12	951.0	1967.0
13	951.0	1940.0
14	934.0	1938.0

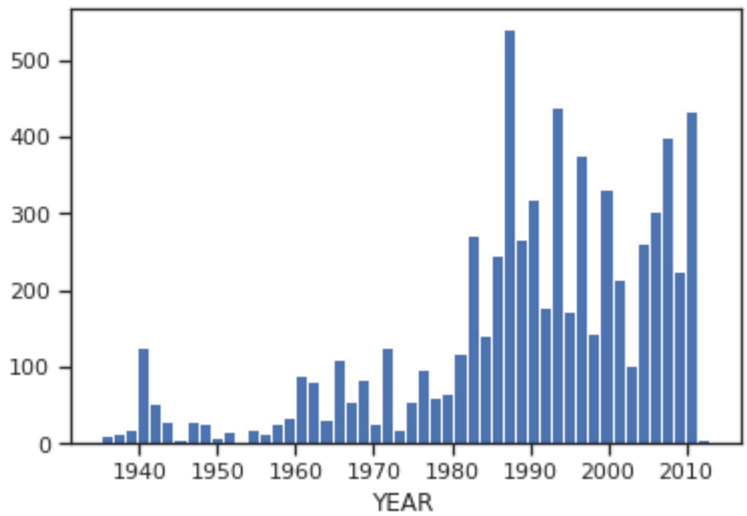
```
15        930.0  1943.0
16        803.0  1940.0
17        716.0  1994.0
18        706.0  1961.0
19        677.0  1986.0
20        654.0  1941.0
21        635.0  1976.0
22        605.0  1942.0
23        595.0  1965.0
24        593.0  1968.0
25        584.0  1980.0
26        560.0  1993.0
27        558.0  1960.0
28        557.0  1986.0
29        549.0  1971.0
...         ...     ...
6866        NaN  1967.0
6867        NaN  1967.0
6868        NaN  1967.0
6869        NaN  1967.0
6870        NaN  1967.0
6871        NaN  1966.0
6872        NaN  1966.0
6873        NaN  1965.0
6874        NaN  1963.0
6875        NaN  1962.0
6876        NaN  1960.0
6877        NaN  1955.0
6878        NaN  1948.0
6879        NaN  1946.0
6880        NaN  1946.0
6881        NaN  1944.0
6882        NaN  1941.0
6883        NaN  1941.0
6884        NaN  1940.0
6885        NaN  1940.0
6886        NaN  1936.0
6887        NaN     NaN
6888        NaN     NaN
6889        NaN     NaN
6890        NaN     NaN
6891        NaN     NaN
6892        NaN     NaN
6893        NaN     NaN
6894        NaN     NaN
6895        NaN     NaN

[6896 rows x 2 columns]
```

In [0]:
```
# Гистограмма по признакам
for col in data_num:
```

```
    plt.hist(data[col], 50)
    plt.xlabel(col)
    plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/numpy/lib/function_base.py:780: RuntimeWarning: inval
  keep = (tmp_a >= first_edge)
/usr/local/lib/python3.6/dist-packages/numpy/lib/function_base.py:781: RuntimeWarning: inval
  keep &= (tmp_a <= last_edge)
```





In [0]:
```
# Фильтр по пустым значениям поля YEAR
data[data['YEAR'].isnull()]
```

Out[0]:
```
     page_id                          name  \
386     1891  Jakeem Williams (New Earth)
```

1400	64303	Hadley Jaggar (New Earth)
1401	13097	Nergal (New Earth)
1832	65286	Gregory Wolfe (New Earth)
1937	146333	Clarence Charles Batson V (New Earth)
1938	113413	Chad Graham (New Earth)
2065	344513	Jupiter (New Earth)
2066	344983	Pegasus (New Earth)
2067	286906	Asteroth (New Earth)
2230	155569	Red Panzer IV (New Earth)
2231	19044	Gernsback (New Earth)
2232	202057	Henry Cosgei (New Earth)
2413	216380	Marilyn Batson (New Earth)
2414	178197	Michael Tree (New Earth)
2841	383108	Brunhilde (New Earth)
2842	251517	Kuan Ti (New Earth)
3104	383914	Helen of Troy (New Earth)
3105	256793	Pluto (New Earth)
3431	15909	Ammon-Ra (New Earth)
3432	348898	Kreaven (New Earth)
3433	345589	Vulcan (New Earth)
3434	57839	Donna Cavanagh (New Earth)
3435	68612	Amadeus Arkham (New Earth)
3819	182833	Scott Spencer (New Earth)
3820	213354	Maria Montez (New Earth)
3821	345591	Diana, Goddess of the Hunt (New Earth)
3822	47346	Gregory the Gargoyle (New Earth)
3823	345586	Minerva (Roman Goddess) (New Earth)
3824	66157	Auerbach (New Earth)
4320	139807	Virgil Adams (New Earth)
...	...	...
5527	112333	Lisa Morice (New Earth)
5528	189975	Carter Nichols (New Earth)
5529	139768	Cupid (New Earth)
5530	345585	Juno (New Earth)
5531	271506	Luki Lo (New Earth)
5532	177249	Stanley Wilson (New Earth)
5533	250224	Elena Leal (New Earth)
5534	185720	Druid (New Earth)
5535	218828	Crone (New Earth)
5536	95738	Fancy Feet (New Earth)
5537	182478	Rico Strada (New Earth)
5538	31642	Benjamin Hubbard (New Earth)
6532	159528	Materna Minnx (New Earth)
6533	19799	Frank Baker, Jr. (New Earth)
6534	242167	Prowley (New Earth)
6535	95767	Smother (New Earth)
6536	16094	Mark Antaeus (New Earth)
6537	128000	Jerome Cox (New Earth)
6538	345590	Apollo (Roman God) (New Earth)
6539	15050	Ben Lo (New Earth)

6540	205584	Auctioneer II (New Earth)
6887	283661	Herbert Hoover (New Earth)
6888	283657	William Howard Taft (New Earth)
6889	21655	Frank Fitzsimmons (New Earth)
6890	283482	James Garfield (New Earth)
6891	66302	Nadine West (New Earth)
6892	283475	Warren Harding (New Earth)
6893	283478	William Harrison (New Earth)
6894	283471	William McKinley (New Earth)
6895	150660	Mookie (New Earth)

	urlslug	ID \
386	\/wiki\/Jakeem_Williams_(New_Earth)	Secret Identity
1400	\/wiki\/Hadley_Jaggar_(New_Earth)	Secret Identity
1401	\/wiki\/Nergal_(New_Earth)	NaN
1832	\/wiki\/Gregory_Wolfe_(New_Earth)	Public Identity
1937	\/wiki\/Clarence_Charles_Batson_V_(New_Earth)	Public Identity
1938	\/wiki\/Chad_Graham_(New_Earth)	Secret Identity
2065	\/wiki\/Jupiter_(New_Earth)	NaN
2066	\/wiki\/Pegasus_(New_Earth)	NaN
2067	\/wiki\/Asteroth_(New_Earth)	Secret Identity
2230	\/wiki\/Red_Panzer_IV_(New_Earth)	Secret Identity
2231	\/wiki\/Gernsback_(New_Earth)	NaN
2232	\/wiki\/Henry_Cosgei_(New_Earth)	Secret Identity
2413	\/wiki\/Marilyn_Batson_(New_Earth)	Public Identity
2414	\/wiki\/Michael_Tree_(New_Earth)	NaN
2841	\/wiki\/Brunhilde_(New_Earth)	NaN
2842	\/wiki\/Kuan_Ti_(New_Earth)	Public Identity
3104	\/wiki\/Helen_of_Troy_(New_Earth)	NaN
3105	\/wiki\/Pluto_(New_Earth)	Public Identity
3431	\/wiki\/Ammon-Ra_(New_Earth)	Secret Identity
3432	\/wiki\/Kreaven_(New_Earth)	Public Identity
3433	\/wiki\/Vulcan_(New_Earth)	NaN
3434	\/wiki\/Donna_Cavanagh_(New_Earth)	Public Identity
3435	\/wiki\/Amadeus_Arkham_(New_Earth)	Public Identity
3819	\/wiki\/Scott_Spencer_(New_Earth)	Public Identity
3820	\/wiki\/Maria_Montez_(New_Earth)	NaN
3821	\/wiki\/Diana,_Goddess_of_the_Hunt_(New_Earth)	NaN
3822	\/wiki\/Gregory_the_Gargoyle_(New_Earth)	NaN
3823	\/wiki\/Minerva_(Roman_Goddess)_(New_Earth)	NaN
3824	\/wiki\/Auerbach_(New_Earth)	Secret Identity
4320	\/wiki\/Virgil_Adams_(New_Earth)	Public Identity
...	...	...
5527	\/wiki\/Lisa_Morice_(New_Earth)	Public Identity
5528	\/wiki\/Carter_Nichols_(New_Earth)	Public Identity
5529	\/wiki\/Cupid_(New_Earth)	NaN
5530	\/wiki\/Juno_(New_Earth)	NaN
5531	\/wiki\/Luki_Lo_(New_Earth)	NaN
5532	\/wiki\/Stanley_Wilson_(New_Earth)	NaN
5533	\/wiki\/Elena_Leal_(New_Earth)	Public Identity

5534	\/wiki\/Druid_(New_Earth)	Secret Identity
5535	\/wiki\/Crone_(New_Earth)	Secret Identity
5536	\/wiki\/Fancy_Feet_(New_Earth)	NaN
5537	\/wiki\/Rico_Strada_(New_Earth)	NaN
5538	\/wiki\/Benjamin_Hubbard_(New_Earth)	NaN
6532	\/wiki\/Materna_Minnx_(New_Earth)	Public Identity
6533	\/wiki\/Frank_Baker,_Jr._(New_Earth)	Public Identity
6534	\/wiki\/Prowley_(New_Earth)	Public Identity
6535	\/wiki\/Smother_(New_Earth)	NaN
6536	\/wiki\/Mark_Antaeus_(New_Earth)	Public Identity
6537	\/wiki\/Jerome_Cox_(New_Earth)	Public Identity
6538	\/wiki\/Apollo_(Roman_God)_(New_Earth)	NaN
6539	\/wiki\/Ben_Lo_(New_Earth)	Public Identity
6540	\/wiki\/Auctioneer_II_(New_Earth)	Secret Identity
6887	\/wiki\/Herbert_Hoover_(New_Earth)	Public Identity
6888	\/wiki\/William_Howard_Taft_(New_Earth)	Public Identity
6889	\/wiki\/Frank_Fitzsimmons_(New_Earth)	Public Identity
6890	\/wiki\/James_Garfield_(New_Earth)	Public Identity
6891	\/wiki\/Nadine_West_(New_Earth)	Public Identity
6892	\/wiki\/Warren_Harding_(New_Earth)	Public Identity
6893	\/wiki\/William_Harrison_(New_Earth)	Public Identity
6894	\/wiki\/William_McKinley_(New_Earth)	Public Identity
6895	\/wiki\/Mookie_(New_Earth)	Public Identity

	ALIGN	EYE	HAIR \
386	NaN	Brown Eyes	NaN
1400	Good Characters	Blue Eyes	Blond Hair
1401	Bad Characters	Yellow Eyes	NaN
1832	Neutral Characters	Brown Eyes	Black Hair
1937	Good Characters	NaN	Black Hair
1938	Bad Characters	NaN	Blond Hair
2065	Good Characters	NaN	White Hair
2066	Good Characters	NaN	Black Hair
2067	Bad Characters	Yellow Eyes	Black Hair
2230	Bad Characters	NaN	NaN
2231	Good Characters	NaN	NaN
2232	Good Characters	Black Eyes	Black Hair
2413	Good Characters	Brown Eyes	Brown Hair
2414	NaN	NaN	Black Hair
2841	Good Characters	NaN	NaN
2842	Good Characters	NaN	Black Hair
3104	Good Characters	NaN	Blond Hair
3105	Bad Characters	NaN	NaN
3431	Neutral Characters	NaN	NaN
3432	Neutral Characters	NaN	NaN
3433	Good Characters	NaN	NaN
3434	Good Characters	Green Eyes	Blond Hair
3435	Good Characters	NaN	NaN
3819	Neutral Characters	NaN	Black Hair
3820	NaN	NaN	NaN

3821	Good Characters	NaN	NaN
3822	Good Characters	NaN	NaN
3823	Good Characters	NaN	NaN
3824	NaN	NaN	Brown Hair
4320	Bad Characters	NaN	Black Hair
...	...	...	...
5527	Good Characters	Grey Eyes	Strawberry Blond Hair
5528	Good Characters	NaN	Brown Hair
5529	Good Characters	NaN	NaN
5530	Good Characters	NaN	NaN
5531	Good Characters	NaN	Black Hair
5532	Good Characters	NaN	Black Hair
5533	Good Characters	NaN	White Hair
5534	Bad Characters	NaN	Black Hair
5535	Bad Characters	NaN	Grey Hair
5536	Bad Characters	NaN	NaN
5537	Bad Characters	Blue Eyes	Blond Hair
5538	NaN	NaN	NaN
6532	NaN	NaN	Brown Hair
6533	Neutral Characters	Blue Eyes	Grey Hair
6534	Neutral Characters	Yellow Eyes	Black Hair
6535	Neutral Characters	NaN	NaN
6536	Good Characters	Blue Eyes	Black Hair
6537	Bad Characters	NaN	NaN
6538	Good Characters	NaN	NaN
6539	Good Characters	Brown Eyes	Black Hair
6540	Bad Characters	NaN	White Hair
6887	Good Characters	NaN	NaN
6888	Good Characters	NaN	NaN
6889	Good Characters	NaN	Grey Hair
6890	Good Characters	NaN	NaN
6891	Good Characters	NaN	NaN
6892	Good Characters	NaN	NaN
6893	Good Characters	NaN	NaN
6894	Good Characters	NaN	NaN
6895	Bad Characters	Blue Eyes	Blond Hair

	SEX	GSM	ALIVE \
386	Male Characters	NaN	Living Characters
1400	Male Characters	NaN	Deceased Characters
1401	Male Characters	NaN	Living Characters
1832	Male Characters	NaN	Living Characters
1937	Male Characters	NaN	Deceased Characters
1938	Male Characters	NaN	Deceased Characters
2065	Male Characters	NaN	Living Characters
2066	Female Characters	NaN	Living Characters
2067	Male Characters	NaN	Living Characters
2230	Male Characters	NaN	Living Characters
2231	Male Characters	NaN	Living Characters
2232	Male Characters	NaN	Living Characters

2413	Female Characters	NaN	Deceased Characters
2414	Female Characters	NaN	Living Characters
2841	Female Characters	NaN	Living Characters
2842	Male Characters	NaN	Living Characters
3104	Female Characters	NaN	Living Characters
3105	Male Characters	NaN	Living Characters
3431	Male Characters	NaN	Living Characters
3432	Male Characters	NaN	Living Characters
3433	Male Characters	NaN	Living Characters
3434	Female Characters	Homosexual Characters	Living Characters
3435	Male Characters	NaN	Living Characters
3819	Male Characters	NaN	Living Characters
3820	Female Characters	NaN	Living Characters
3821	Female Characters	NaN	Living Characters
3822	Male Characters	NaN	Living Characters
3823	Female Characters	NaN	Living Characters
3824	Male Characters	NaN	Living Characters
4320	Male Characters	NaN	Living Characters
...	...	...	...
5527	Female Characters	NaN	Living Characters
5528	Male Characters	NaN	Deceased Characters
5529	Male Characters	NaN	Living Characters
5530	Female Characters	NaN	Living Characters
5531	Male Characters	NaN	Living Characters
5532	Male Characters	NaN	Living Characters
5533	Female Characters	NaN	Living Characters
5534	Male Characters	NaN	Living Characters
5535	Female Characters	NaN	Living Characters
5536	Male Characters	NaN	Living Characters
5537	Male Characters	NaN	Living Characters
5538	Male Characters	NaN	Living Characters
6532	Female Characters	NaN	Living Characters
6533	Male Characters	NaN	Living Characters
6534	NaN	NaN	Living Characters
6535	Female Characters	NaN	Living Characters
6536	Male Characters	NaN	Deceased Characters
6537	Male Characters	NaN	Living Characters
6538	Male Characters	NaN	Living Characters
6539	Male Characters	NaN	Living Characters
6540	Male Characters	NaN	Living Characters
6887	Male Characters	NaN	Living Characters
6888	Male Characters	NaN	Living Characters
6889	Male Characters	NaN	Living Characters
6890	Male Characters	NaN	Living Characters
6891	Female Characters	NaN	Living Characters
6892	Male Characters	NaN	Living Characters
6893	Male Characters	NaN	Living Characters
6894	Male Characters	NaN	Living Characters
6895	Male Characters	NaN	Living Characters

	APPEARANCES	FIRST APPEARANCE	YEAR
386	79.0	NaN	NaN
1400	19.0	NaN	NaN
1401	19.0	NaN	NaN
1832	14.0	NaN	NaN
1937	13.0	NaN	NaN
1938	13.0	NaN	NaN
2065	12.0	NaN	NaN
2066	12.0	NaN	NaN
2067	12.0	NaN	NaN
2230	11.0	NaN	NaN
2231	11.0	NaN	NaN
2232	11.0	NaN	NaN
2413	10.0	NaN	NaN
2414	10.0	NaN	NaN
2841	8.0	NaN	NaN
2842	8.0	NaN	NaN
3104	7.0	NaN	NaN
3105	7.0	NaN	NaN
3431	6.0	NaN	NaN
3432	6.0	NaN	NaN
3433	6.0	NaN	NaN
3434	6.0	NaN	NaN
3435	6.0	NaN	NaN
3819	5.0	NaN	NaN
3820	5.0	NaN	NaN
3821	5.0	NaN	NaN
3822	5.0	NaN	NaN
3823	5.0	NaN	NaN
3824	5.0	NaN	NaN
4320	4.0	NaN	NaN
...	...	...	...
5527	2.0	NaN	NaN
5528	2.0	NaN	NaN
5529	2.0	NaN	NaN
5530	2.0	NaN	NaN
5531	2.0	NaN	NaN
5532	2.0	NaN	NaN
5533	2.0	NaN	NaN
5534	2.0	NaN	NaN
5535	2.0	NaN	NaN
5536	2.0	NaN	NaN
5537	2.0	NaN	NaN
5538	2.0	NaN	NaN
6532	1.0	NaN	NaN
6533	1.0	NaN	NaN
6534	1.0	NaN	NaN
6535	1.0	NaN	NaN
6536	1.0	NaN	NaN
6537	1.0	NaN	NaN

```
6538          1.0           NaN   NaN
6539          1.0           NaN   NaN
6540          1.0           NaN   NaN
6887          NaN             NaN   NaN
6888          NaN             NaN   NaN
6889          NaN             NaN   NaN
6890          NaN             NaN   NaN
6891          NaN             NaN   NaN
6892          NaN             NaN   NaN
6893          NaN             NaN   NaN
6894          NaN             NaN   NaN
6895          NaN             NaN   NaN

[69 rows x 13 columns]
```

In [0]:
```
# Запоминаем индексы строк с пустыми значениями
flt_index = data[data['YEAR'].isnull()].index
flt_index
```

Out[0]:
```
Int64Index([ 386, 1400, 1401, 1832, 1937, 1938, 2065, 2066, 2067, 2230, 2231,
            2232, 2413, 2414, 2841, 2842, 3104, 3105, 3431, 3432, 3433, 3434,
            3435, 3819, 3820, 3821, 3822, 3823, 3824, 4320, 4321, 4322, 4323,
            4826, 4827, 4828, 4829, 5525, 5526, 5527, 5528, 5529, 5530, 5531,
            5532, 5533, 5534, 5535, 5536, 5537, 5538, 6532, 6533, 6534, 6535,
            6536, 6537, 6538, 6539, 6540, 6887, 6888, 6889, 6890, 6891, 6892,
            6893, 6894, 6895],
           dtype='int64')
```

In [0]:
```
# Проверяем что выводятся нужные строки
data[data.index.isin(flt_index)]
```

Out[0]:
```
      page_id                                  name  \
386      1891          Jakeem Williams (New Earth)
1400    64303            Hadley Jaggar (New Earth)
1401    13097                   Nergal (New Earth)
1832    65286            Gregory Wolfe (New Earth)
1937   146333  Clarence Charles Batson V (New Earth)
1938   113413             Chad Graham (New Earth)
2065   344513                 Jupiter (New Earth)
2066   344983                 Pegasus (New Earth)
2067   286906                Asteroth (New Earth)
2230   155569           Red Panzer IV (New Earth)
2231    19044              Gernsback (New Earth)
2232   202057            Henry Cosgei (New Earth)
2413   216380          Marilyn Batson (New Earth)
2414   178197            Michael Tree (New Earth)
2841   383108               Brunhilde (New Earth)
2842   251517                 Kuan Ti (New Earth)
3104   383914            Helen of Troy (New Earth)
3105   256793                   Pluto (New Earth)
3431    15909                Ammon-Ra (New Earth)
```

3432	348898	Kreaven (New Earth)
3433	345589	Vulcan (New Earth)
3434	57839	Donna Cavanagh (New Earth)
3435	68612	Amadeus Arkham (New Earth)
3819	182833	Scott Spencer (New Earth)
3820	213354	Maria Montez (New Earth)
3821	345591	Diana, Goddess of the Hunt (New Earth)
3822	47346	Gregory the Gargoyle (New Earth)
3823	345586	Minerva (Roman Goddess) (New Earth)
3824	66157	Auerbach (New Earth)
4320	139807	Virgil Adams (New Earth)
...	...	...
5527	112333	Lisa Morice (New Earth)
5528	189975	Carter Nichols (New Earth)
5529	139768	Cupid (New Earth)
5530	345585	Juno (New Earth)
5531	271506	Luki Lo (New Earth)
5532	177249	Stanley Wilson (New Earth)
5533	250224	Elena Leal (New Earth)
5534	185720	Druid (New Earth)
5535	218828	Crone (New Earth)
5536	95738	Fancy Feet (New Earth)
5537	182478	Rico Strada (New Earth)
5538	31642	Benjamin Hubbard (New Earth)
6532	159528	Materna Minnx (New Earth)
6533	19799	Frank Baker, Jr. (New Earth)
6534	242167	Prowley (New Earth)
6535	95767	Smother (New Earth)
6536	16094	Mark Antaeus (New Earth)
6537	128000	Jerome Cox (New Earth)
6538	345590	Apollo (Roman God) (New Earth)
6539	15050	Ben Lo (New Earth)
6540	205584	Auctioneer II (New Earth)
6887	283661	Herbert Hoover (New Earth)
6888	283657	William Howard Taft (New Earth)
6889	21655	Frank Fitzsimmons (New Earth)
6890	283482	James Garfield (New Earth)
6891	66302	Nadine West (New Earth)
6892	283475	Warren Harding (New Earth)
6893	283478	William Harrison (New Earth)
6894	283471	William McKinley (New Earth)
6895	150660	Mookie (New Earth)

	urlslug	ID \
386	\/wiki\/Jakeem_Williams_(New_Earth)	Secret Identity
1400	\/wiki\/Hadley_Jaggar_(New_Earth)	Secret Identity
1401	\/wiki\/Nergal_(New_Earth)	NaN
1832	\/wiki\/Gregory_Wolfe_(New_Earth)	Public Identity
1937	\/wiki\/Clarence_Charles_Batson_V_(New_Earth)	Public Identity
1938	\/wiki\/Chad_Graham_(New_Earth)	Secret Identity

2065	\/wiki\/Jupiter_(New_Earth)	NaN
2066	\/wiki\/Pegasus_(New_Earth)	NaN
2067	\/wiki\/Asteroth_(New_Earth)	Secret Identity
2230	\/wiki\/Red_Panzer_IV_(New_Earth)	Secret Identity
2231	\/wiki\/Gernsback_(New_Earth)	NaN
2232	\/wiki\/Henry_Cosgei_(New_Earth)	Secret Identity
2413	\/wiki\/Marilyn_Batson_(New_Earth)	Public Identity
2414	\/wiki\/Michael_Tree_(New_Earth)	NaN
2841	\/wiki\/Brunhilde_(New_Earth)	NaN
2842	\/wiki\/Kuan_Ti_(New_Earth)	Public Identity
3104	\/wiki\/Helen_of_Troy_(New_Earth)	NaN
3105	\/wiki\/Pluto_(New_Earth)	Public Identity
3431	\/wiki\/Ammon-Ra_(New_Earth)	Secret Identity
3432	\/wiki\/Kreaven_(New_Earth)	Public Identity
3433	\/wiki\/Vulcan_(New_Earth)	NaN
3434	\/wiki\/Donna_Cavanagh_(New_Earth)	Public Identity
3435	\/wiki\/Amadeus_Arkham_(New_Earth)	Public Identity
3819	\/wiki\/Scott_Spencer_(New_Earth)	Public Identity
3820	\/wiki\/Maria_Montez_(New_Earth)	NaN
3821	\/wiki\/Diana,_Goddess_of_the_Hunt_(New_Earth)	NaN
3822	\/wiki\/Gregory_the_Gargoyle_(New_Earth)	NaN
3823	\/wiki\/Minerva_(Roman_Goddess)_(New_Earth)	NaN
3824	\/wiki\/Auerbach_(New_Earth)	Secret Identity
4320	\/wiki\/Virgil_Adams_(New_Earth)	Public Identity
...	...	...
5527	\/wiki\/Lisa_Morice_(New_Earth)	Public Identity
5528	\/wiki\/Carter_Nichols_(New_Earth)	Public Identity
5529	\/wiki\/Cupid_(New_Earth)	NaN
5530	\/wiki\/Juno_(New_Earth)	NaN
5531	\/wiki\/Luki_Lo_(New_Earth)	NaN
5532	\/wiki\/Stanley_Wilson_(New_Earth)	NaN
5533	\/wiki\/Elena_Leal_(New_Earth)	Public Identity
5534	\/wiki\/Druid_(New_Earth)	Secret Identity
5535	\/wiki\/Crone_(New_Earth)	Secret Identity
5536	\/wiki\/Fancy_Feet_(New_Earth)	NaN
5537	\/wiki\/Rico_Strada_(New_Earth)	NaN
5538	\/wiki\/Benjamin_Hubbard_(New_Earth)	NaN
6532	\/wiki\/Materna_Minnx_(New_Earth)	Public Identity
6533	\/wiki\/Frank_Baker,_Jr._(New_Earth)	Public Identity
6534	\/wiki\/Prowley_(New_Earth)	Public Identity
6535	\/wiki\/Smother_(New_Earth)	NaN
6536	\/wiki\/Mark_Antaeus_(New_Earth)	Public Identity
6537	\/wiki\/Jerome_Cox_(New_Earth)	Public Identity
6538	\/wiki\/Apollo_(Roman_God)_(New_Earth)	NaN
6539	\/wiki\/Ben_Lo_(New_Earth)	Public Identity
6540	\/wiki\/Auctioneer_II_(New_Earth)	Secret Identity
6887	\/wiki\/Herbert_Hoover_(New_Earth)	Public Identity
6888	\/wiki\/William_Howard_Taft_(New_Earth)	Public Identity
6889	\/wiki\/Frank_Fitzsimmons_(New_Earth)	Public Identity
6890	\/wiki\/James_Garfield_(New_Earth)	Public Identity

6891	\/wiki\/Nadine_West_(New_Earth)	Public Identity
6892	\/wiki\/Warren_Harding_(New_Earth)	Public Identity
6893	\/wiki\/William_Harrison_(New_Earth)	Public Identity
6894	\/wiki\/William_McKinley_(New_Earth)	Public Identity
6895	\/wiki\/Mookie_(New_Earth)	Public Identity

	ALIGN	EYE	HAIR \
386	NaN	Brown Eyes	NaN
1400	Good Characters	Blue Eyes	Blond Hair
1401	Bad Characters	Yellow Eyes	NaN
1832	Neutral Characters	Brown Eyes	Black Hair
1937	Good Characters	NaN	Black Hair
1938	Bad Characters	NaN	Blond Hair
2065	Good Characters	NaN	White Hair
2066	Good Characters	NaN	Black Hair
2067	Bad Characters	Yellow Eyes	Black Hair
2230	Bad Characters	NaN	NaN
2231	Good Characters	NaN	NaN
2232	Good Characters	Black Eyes	Black Hair
2413	Good Characters	Brown Eyes	Brown Hair
2414	NaN	NaN	Black Hair
2841	Good Characters	NaN	NaN
2842	Good Characters	NaN	Black Hair
3104	Good Characters	NaN	Blond Hair
3105	Bad Characters	NaN	NaN
3431	Neutral Characters	NaN	NaN
3432	Neutral Characters	NaN	NaN
3433	Good Characters	NaN	NaN
3434	Good Characters	Green Eyes	Blond Hair
3435	Good Characters	NaN	NaN
3819	Neutral Characters	NaN	Black Hair
3820	NaN	NaN	NaN
3821	Good Characters	NaN	NaN
3822	Good Characters	NaN	NaN
3823	Good Characters	NaN	NaN
3824	NaN	NaN	Brown Hair
4320	Bad Characters	NaN	Black Hair
...	...	...	...
5527	Good Characters	Grey Eyes	Strawberry Blond Hair
5528	Good Characters	NaN	Brown Hair
5529	Good Characters	NaN	NaN
5530	Good Characters	NaN	NaN
5531	Good Characters	NaN	Black Hair
5532	Good Characters	NaN	Black Hair
5533	Good Characters	NaN	White Hair
5534	Bad Characters	NaN	Black Hair
5535	Bad Characters	NaN	Grey Hair
5536	Bad Characters	NaN	NaN
5537	Bad Characters	Blue Eyes	Blond Hair
5538	NaN	NaN	NaN

6532	NaN	NaN	Brown Hair
6533	Neutral Characters	Blue Eyes	Grey Hair
6534	Neutral Characters	Yellow Eyes	Black Hair
6535	Neutral Characters	NaN	NaN
6536	Good Characters	Blue Eyes	Black Hair
6537	Bad Characters	NaN	NaN
6538	Good Characters	NaN	NaN
6539	Good Characters	Brown Eyes	Black Hair
6540	Bad Characters	NaN	White Hair
6887	Good Characters	NaN	NaN
6888	Good Characters	NaN	NaN
6889	Good Characters	NaN	Grey Hair
6890	Good Characters	NaN	NaN
6891	Good Characters	NaN	NaN
6892	Good Characters	NaN	NaN
6893	Good Characters	NaN	NaN
6894	Good Characters	NaN	NaN
6895	Bad Characters	Blue Eyes	Blond Hair

	SEX	GSM	ALIVE \
386	Male Characters	NaN	Living Characters
1400	Male Characters	NaN	Deceased Characters
1401	Male Characters	NaN	Living Characters
1832	Male Characters	NaN	Living Characters
1937	Male Characters	NaN	Deceased Characters
1938	Male Characters	NaN	Deceased Characters
2065	Male Characters	NaN	Living Characters
2066	Female Characters	NaN	Living Characters
2067	Male Characters	NaN	Living Characters
2230	Male Characters	NaN	Living Characters
2231	Male Characters	NaN	Living Characters
2232	Male Characters	NaN	Living Characters
2413	Female Characters	NaN	Deceased Characters
2414	Female Characters	NaN	Living Characters
2841	Female Characters	NaN	Living Characters
2842	Male Characters	NaN	Living Characters
3104	Female Characters	NaN	Living Characters
3105	Male Characters	NaN	Living Characters
3431	Male Characters	NaN	Living Characters
3432	Male Characters	NaN	Living Characters
3433	Male Characters	NaN	Living Characters
3434	Female Characters	Homosexual Characters	Living Characters
3435	Male Characters	NaN	Living Characters
3819	Male Characters	NaN	Living Characters
3820	Female Characters	NaN	Living Characters
3821	Female Characters	NaN	Living Characters
3822	Male Characters	NaN	Living Characters
3823	Female Characters	NaN	Living Characters
3824	Male Characters	NaN	Living Characters
4320	Male Characters	NaN	Living Characters

...	...	...	...
5527	Female Characters	NaN	Living Characters
5528	Male Characters	NaN	Deceased Characters
5529	Male Characters	NaN	Living Characters
5530	Female Characters	NaN	Living Characters
5531	Male Characters	NaN	Living Characters
5532	Male Characters	NaN	Living Characters
5533	Female Characters	NaN	Living Characters
5534	Male Characters	NaN	Living Characters
5535	Female Characters	NaN	Living Characters
5536	Male Characters	NaN	Living Characters
5537	Male Characters	NaN	Living Characters
5538	Male Characters	NaN	Living Characters
6532	Female Characters	NaN	Living Characters
6533	Male Characters	NaN	Living Characters
6534	NaN	NaN	Living Characters
6535	Female Characters	NaN	Living Characters
6536	Male Characters	NaN	Deceased Characters
6537	Male Characters	NaN	Living Characters
6538	Male Characters	NaN	Living Characters
6539	Male Characters	NaN	Living Characters
6540	Male Characters	NaN	Living Characters
6887	Male Characters	NaN	Living Characters
6888	Male Characters	NaN	Living Characters
6889	Male Characters	NaN	Living Characters
6890	Male Characters	NaN	Living Characters
6891	Female Characters	NaN	Living Characters
6892	Male Characters	NaN	Living Characters
6893	Male Characters	NaN	Living Characters
6894	Male Characters	NaN	Living Characters
6895	Male Characters	NaN	Living Characters

	APPEARANCES	FIRST APPEARANCE	YEAR
386	79.0	NaN	NaN
1400	19.0	NaN	NaN
1401	19.0	NaN	NaN
1832	14.0	NaN	NaN
1937	13.0	NaN	NaN
1938	13.0	NaN	NaN
2065	12.0	NaN	NaN
2066	12.0	NaN	NaN
2067	12.0	NaN	NaN
2230	11.0	NaN	NaN
2231	11.0	NaN	NaN
2232	11.0	NaN	NaN
2413	10.0	NaN	NaN
2414	10.0	NaN	NaN
2841	8.0	NaN	NaN
2842	8.0	NaN	NaN
3104	7.0	NaN	NaN

```
3105      7.0        NaN  NaN
3431      6.0        NaN  NaN
3432      6.0        NaN  NaN
3433      6.0        NaN  NaN
3434      6.0        NaN  NaN
3435      6.0        NaN  NaN
3819      5.0        NaN  NaN
3820      5.0        NaN  NaN
3821      5.0        NaN  NaN
3822      5.0        NaN  NaN
3823      5.0        NaN  NaN
3824      5.0        NaN  NaN
4320      4.0        NaN  NaN
...     ...       ...  ...
5527      2.0        NaN  NaN
5528      2.0        NaN  NaN
5529      2.0        NaN  NaN
5530      2.0        NaN  NaN
5531      2.0        NaN  NaN
5532      2.0        NaN  NaN
5533      2.0        NaN  NaN
5534      2.0        NaN  NaN
5535      2.0        NaN  NaN
5536      2.0        NaN  NaN
5537      2.0        NaN  NaN
5538      2.0        NaN  NaN
6532      1.0        NaN  NaN
6533      1.0        NaN  NaN
6534      1.0        NaN  NaN
6535      1.0        NaN  NaN
6536      1.0        NaN  NaN
6537      1.0        NaN  NaN
6538      1.0        NaN  NaN
6539      1.0        NaN  NaN
6540      1.0        NaN  NaN
6887      NaN         NaN  NaN
6888      NaN         NaN  NaN
6889      NaN         NaN  NaN
6890      NaN         NaN  NaN
6891      NaN         NaN  NaN
6892      NaN         NaN  NaN
6893      NaN         NaN  NaN
6894      NaN         NaN  NaN
6895      NaN         NaN  NaN

[69 rows x 13 columns]
```

In [0]:
```
data_num_Year = data_num[['YEAR']]
data_num_Year.head()
```

Out[0]:
```
    YEAR
```

```
0   1939.0
1   1986.0
2   1959.0
3   1987.0
4   1940.0
```

```
In [0]: from sklearn.impute import SimpleImputer
        from sklearn.impute import MissingIndicator
```

```
In [0]: # Фильтр для проверки заполнения пустых значений
        indicator = MissingIndicator()
        mask_missing_values_only = indicator.fit_transform(data_num_Year)
        mask_missing_values_only
```

```
Out[0]: array([[False],
               [False],
               [False],
               ...,
               [ True],
               [ True],
               [ True]])
```

```
In [0]: strategies=['mean', 'median','most_frequent']
```

```
In [0]: def test_num_impute(strategy_param):
            imp_num = SimpleImputer(strategy=strategy_param)
            data_num_imp = imp_num.fit_transform(data_num_Year)
            return data_num_imp[mask_missing_values_only]
```

```
In [0]: strategies[0], test_num_impute(strategies[0])
```

```
Out[0]: ('mean', array([1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178, 1989.76666178, 1989.76666178, 1989.76666178,
                1989.76666178]))
```

```
In [0]: strategies[1], test_num_impute(strategies[1])
```

Out[0]: ('median',
```
array([1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992.,
       1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992.,
       1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992.,
       1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992.,
       1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992.,
       1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992.,
       1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992., 1992.,
       1992., 1992., 1992., 1992., 1992., 1992.]))
```

In [0]:
```
strategies[2], test_num_impute(strategies[2])
```

Out[0]: ('most_frequent',
```
array([2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006.,
       2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006.,
       2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006.,
       2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006.,
       2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006.,
       2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006.,
       2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006., 2006.,
       2006., 2006., 2006., 2006., 2006., 2006.]))
```

2.4. Внедрение значений (категориальные данные)

In [0]:
```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
cat_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'
              .format(col, dt, temp_null_count, temp_perc))
```

```
Колонка ID. Тип данных object. Количество пустых значений 2013, 29.19%.
Колонка ALIGN. Тип данных object. Количество пустых значений 601, 8.72%.
Колонка EYE. Тип данных object. Количество пустых значений 3628, 52.61%.
Колонка HAIR. Тип данных object. Количество пустых значений 2274, 32.98%.
Колонка SEX. Тип данных object. Количество пустых значений 125, 1.81%.
Колонка GSM. Тип данных object. Количество пустых значений 6832, 99.07%.
Колонка ALIVE. Тип данных object. Количество пустых значений 3, 0.04%.
Колонка FIRST APPEARANCE. Тип данных object. Количество пустых значений 69, 1.0%.
```

In [0]:
```
cat_temp_data = data[['HAIR']]
cat_temp_data.tail(10)
```

Out[0]:

```
     HAIR
6886  NaN
```

```
6887          NaN
6888          NaN
6889    Grey Hair
6890          NaN
6891          NaN
6892          NaN
6893          NaN
6894          NaN
6895   Blond Hair
```

```
In [0]: cat_temp_data['HAIR'].unique()
```

```
Out[0]: array(['Black Hair', 'Brown Hair', 'White Hair', 'Blond Hair', 'Red Hair',
               nan, 'Green Hair', 'Strawberry Blond Hair', 'Grey Hair',
               'Silver Hair', 'Orange Hair', 'Purple Hair', 'Gold Hair',
               'Blue Hair', 'Reddish Brown Hair', 'Pink Hair', 'Violet Hair',
               'Platinum Blond Hair'], dtype=object)
```

```
In [0]: cat_temp_data[cat_temp_data['HAIR'].isnull()].shape
```

```
Out[0]: (2274, 1)
```

```
In [0]: # Импьютация наиболее частыми значениями
        imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
        data_imp2 = imp2.fit_transform(cat_temp_data)
        data_imp2
```

```
Out[0]: array([['Black Hair'],
               ['Black Hair'],
               ['Brown Hair'],
               ...,
               ['Black Hair'],
               ['Black Hair'],
               ['Blond Hair']], dtype=object)
```

```
In [0]: # Пустые значения отсутствуют
        np.unique(data_imp2)
```

```
Out[0]: array(['Black Hair', 'Blond Hair', 'Blue Hair', 'Brown Hair', 'Gold Hair',
               'Green Hair', 'Grey Hair', 'Orange Hair', 'Pink Hair',
               'Platinum Blond Hair', 'Purple Hair', 'Red Hair',
               'Reddish Brown Hair', 'Silver Hair', 'Strawberry Blond Hair',
               'Violet Hair', 'White Hair'], dtype=object)
```

2.5. Преобразование категориальных признаков в числовые

```
In [0]: cat_enc = pd.DataFrame({'c1':data_imp2.T[0]})
        cat_enc
```

```
Out[0]:          c1
        0    Black Hair
        1    Black Hair
```

2	Brown Hair
3	White Hair
4	Black Hair
5	Black Hair
6	Blond Hair
7	Black Hair
8	Blond Hair
9	Blond Hair
10	Blond Hair
11	Blond Hair
12	Red Hair
13	Brown Hair
14	Black Hair
15	Black Hair
16	Brown Hair
17	Black Hair
18	Black Hair
19	Black Hair
20	Red Hair
21	Blond Hair
22	Black Hair
23	Green Hair
24	Red Hair
25	Black Hair
26	Black Hair
27	Red Hair
28	Red Hair
29	Black Hair
...	...
6866	Black Hair
6867	Black Hair
6868	Black Hair
6869	Black Hair
6870	Black Hair
6871	Black Hair
6872	Black Hair
6873	Black Hair
6874	Black Hair
6875	Black Hair
6876	Red Hair
6877	Black Hair
6878	Blond Hair
6879	Black Hair
6880	Black Hair
6881	Red Hair
6882	Brown Hair
6883	Black Hair
6884	Black Hair
6885	Black Hair
6886	Black Hair

```
6887  Black Hair
6888  Black Hair
6889   Grey Hair
6890  Black Hair
6891  Black Hair
6892  Black Hair
6893  Black Hair
6894  Black Hair
6895  Blond Hair

[6896 rows x 1 columns]
```

2.5.1. Label encoding

In [0]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

In [0]:
```
le = LabelEncoder()
cat_enc_le = le.fit_transform(cat_enc['c1'])
```

In [0]:
```
cat_enc['c1'].unique()
```

Out[0]:
```
array(['Black Hair', 'Brown Hair', 'White Hair', 'Blond Hair', 'Red Hair',
       'Green Hair', 'Strawberry Blond Hair', 'Grey Hair', 'Silver Hair',
       'Orange Hair', 'Purple Hair', 'Gold Hair', 'Blue Hair',
       'Reddish Brown Hair', 'Pink Hair', 'Violet Hair',
       'Platinum Blond Hair'], dtype=object)
```

In [0]:
```
np.unique(cat_enc_le)
```

Out[0]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16])
```

In [0]:
```
le.inverse_transform([x for x in range(16)])
```

Out[0]:
```
array(['Black Hair', 'Blond Hair', 'Blue Hair', 'Brown Hair', 'Gold Hair',
       'Green Hair', 'Grey Hair', 'Orange Hair', 'Pink Hair',
       'Platinum Blond Hair', 'Purple Hair', 'Red Hair',
       'Reddish Brown Hair', 'Silver Hair', 'Strawberry Blond Hair',
       'Violet Hair'], dtype=object)
```

In [0]:
```
cat_enc_le
```

Out[0]:
```
array([0, 0, 3, ..., 0, 0, 1])
```

2.5.2. One-hot encoding

In [0]:
```
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc[['c1']])
```

In [0]:
```
cat_enc.shape
```

Out[0]:
```
(6896, 1)
```

In [0]:
```
cat_enc_ohe.shape
```

Out[0]: (6896, 17)

In [0]: cat_enc_ohe

Out[0]: <6896x17 sparse matrix of type '<class 'numpy.float64'>'
	with 6896 stored elements in Compressed Sparse Row format>

In [0]: cat_enc_ohe.todense()[0:10]

```
Out[0]: matrix([[1., 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.,
          0.],
         [0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
          0.],
         [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
          1.],
         [1., 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.,
          0.],
         [0., 1., 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.,
          0.],
         [0., 1., 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.]])
```

2.5.3. Или

In [0]: pd.get_dummies(cat_enc).tail()

Out[0]:

	c1_Black Hair	c1_Blond Hair	c1_Blue Hair	c1_Brown Hair	c1_Gold Hair	\
6891	1	0	0	0	0	
6892	1	0	0	0	0	
6893	1	0	0	0	0	
6894	1	0	0	0	0	
6895	0	1	0	0	0	

	c1_Green Hair	c1_Grey Hair	c1_Orange Hair	c1_Pink Hair	\
6891	0	0	0	0	
6892	0	0	0	0	
6893	0	0	0	0	
6894	0	0	0	0	
6895	0	0	0	0	

	c1_Platinum Blond Hair	c1_Purple Hair	c1_Red Hair	\
6891	0	0	0	
6892	0	0	0	

6893	0	0	0
6894	0	0	0
6895	0	0	0

	c1_Reddish Brown Hair	c1_Silver Hair	c1_Strawberry Blond Hair \
6891	0	0	0
6892	0	0	0
6893	0	0	0
6894	0	0	0
6895	0	0	0

	c1_Violet Hair	c1_White Hair
6891	0	0
6892	0	0
6893	0	0
6894	0	0
6895	0	0

In [0]:
```
pd.get_dummies(cat_temp_data, dummy_na=True).tail()
```

Out[0]:

	HAIR_Black Hair	HAIR_Blond Hair	HAIR_Blue Hair	HAIR_Brown Hair \
6891	0	0	0	0
6892	0	0	0	0
6893	0	0	0	0
6894	0	0	0	0
6895	0	1	0	0

	HAIR_Gold Hair	HAIR_Green Hair	HAIR_Grey Hair	HAIR_Orange Hair \
6891	0	0	0	0
6892	0	0	0	0
6893	0	0	0	0
6894	0	0	0	0
6895	0	0	0	0

	HAIR_Pink Hair	HAIR_Platinum Blond Hair	HAIR_Purple Hair \
6891	0	0	0
6892	0	0	0
6893	0	0	0
6894	0	0	0
6895	0	0	0

	HAIR_Red Hair	HAIR_Reddish Brown Hair	HAIR_Silver Hair \
6891	0	0	0
6892	0	0	0
6893	0	0	0
6894	0	0	0
6895	0	0	0

	HAIR_Strawberry Blond Hair	HAIR_Violet Hair	HAIR_White Hair	HAIR_nan
6891	0	0	0	1
6892	0	0	0	1

6893	0	0	0	1
6894	0	0	0	1
6895	0	0	0	0

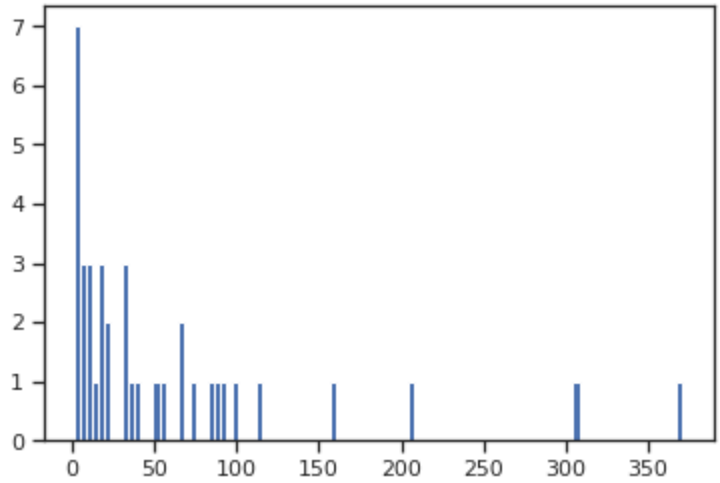
2.6. Масштабирование данных

```
In [0]: from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

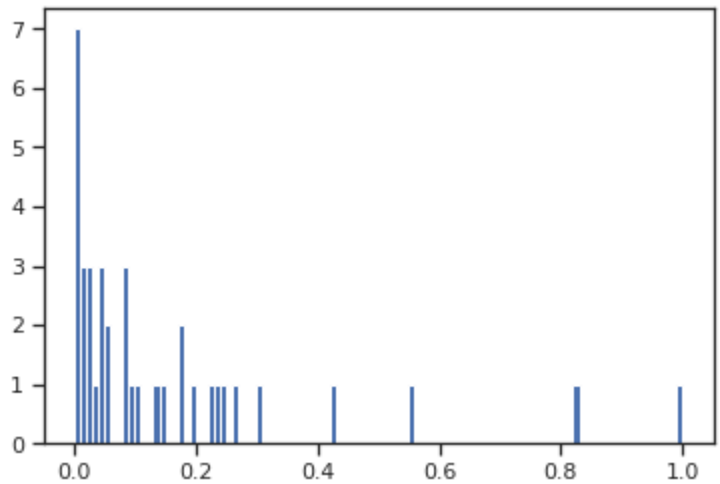
2.6.1. MinMax масштабирование

```
In [0]: sc1 = MinMaxScaler()
        sc1_data = sc1.fit_transform(data_new_2[['APPEARANCES']])
```

```
In [0]: plt.hist(data_new_2['APPEARANCES'], 100)
        plt.show()
```



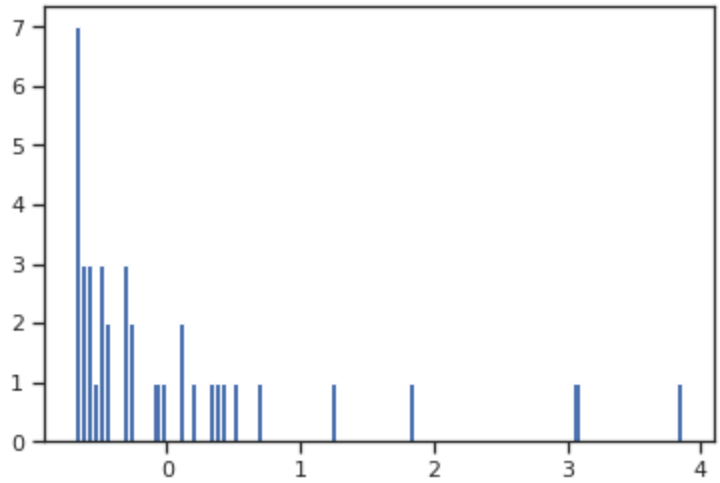

```
In [0]: plt.hist(sc1_data, 100)
        plt.show()
```



```
In [0]: sc2 = StandardScaler()
        sc2_data = sc2.fit_transform(data_new_2[['APPEARANCES']])
```

2.6.2. Z-оценка

```
In [0]: plt.hist(sc2_data, 100)
        plt.show()
```



2.6.3. Нормализация данных

```
In [0]: sc3 = Normalizer()
        sc3_data = sc3.fit_transform(data_new_2[['APPEARANCES']])
```

```
In [0]: plt.hist(sc3_data, 50)
        plt.show()
```