Рубежный контроль №1

Пенегина Вероника ИУ5-65Б

11 Вариант

Задание:

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

Импорт библиотек:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
sns.set (style="ticks")
```

```
data = pd.read_csv('marvel-wikia.csv', sep=',')
```

Характеристика датасета:

```
data.head()
```

]:

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	AP
0	1678	Spider-Man (Peter Parker)	\/Spider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	4043.0	
1	7139	Captain America (Steven Rogers)	\/Captain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	3360.0	
2	64786	Wolverine (James \"Logan\" Howlett)	\/Wolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	3061.0	
3	1868	Iron Man (Anthony \"Tony\" Stark)	\/Iron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	2961.0	
4	2460	Thor (Thor Odinson)	\/Thor_(Thor_Odinson)	No Dual Identity	Good Characters	Blue Eyes	Blond Hair	Male Characters	NaN	Living Characters	2258.0	

<

```
In [8]: # Колонки с пропусками
cols_with_na = [c for c in data.columns if data[c].isnull().sum() > 0]
cols_with_na
```

```
Out[8]: ['ID',
 'ALIGN',
 'EYE',
 'HAIR',
 'SEX',
 'GSM',
 'ALIVE',
 'APPEARANCES',
 'FIRST APPEARANCE',
 'Year']
```

In [9]:
```
# доля (процент) пропусков
[(c, data[c].isnull().mean()) for c in cols_with_na]
```

Out[9]:
```
[('ID', 0.23021494870542256),
 ('ALIGN', 0.17171470444553005),
 ('EYE', 0.5964215925744992),
 ('HAIR', 0.26038104543234003),
 ('SEX', 0.052149487054225695),
 ('GSM', 0.9945041524181729),
 ('ALIVE', 0.00018319491939423546),
 ('APPEARANCES', 0.06692721055202736),
 ('FIRST APPEARANCE', 0.04976795310210064),
 ('Year', 0.04976795310210064)]
```

Обработка пропусков для категориального признака

Можно заметить,что для признака "GSM" пропущенных данных слишком много (около 99%), следовательно нужно удалить признак (колонку) целиком.

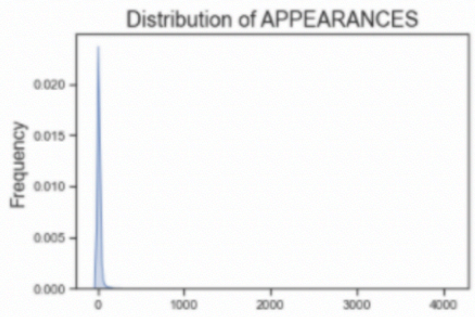
In [10]:
```
data.drop(['GSM'], axis=1, inplace=True)
```

Обработка пропусков для количественного признака

Поскольку в исследуемом датасете один количественный признак "Appearances" и процент пропусков для него составляет <5%, то будем использовать метод заполнения пропущенных значений показателями центра распределения.

In [22]:
```
g = sns.kdeplot(data=data, x="APPEARANCES", shade=True)
g.set_xlabel("APPEARANCES", size = 16)
g.set_ylabel("Frequency", size = 16)
plt.title('Distribution of APPEARANCES', size = 18)
```

Out[22]:
```
Text(0.5, 1.0, 'Distribution of APPEARANCES')
```



In [23]:
```
data[['APPEARANCES']].describe()
```

Out[23]:

	APPEARANCES
count	15280.000000
mean	17.033377
std	96.372959
min	1.000000
25%	1.000000
50%	3.000000
75%	8.000000
max	4043.000000

Получаем одномодальное распределение, поэтому будем использовать моду для заполнения пустых значений.

In [24]:
```
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data[['APPEARANCES']])
imp_num = SimpleImputer(strategy='most_frequent')
data_num_imp = imp_num.fit_transform(data[['APPEARANCES']])
data['APPEARANCES'] =data_num_imp
```

Характеристики датасета после обработки пропусков

In [25]:
```
# Колонки с пропусками
cols_with_na = [c for c in data.columns if data[c].isnull().sum() > 0]
cols_with_na
```

Out[25]:
```
['ID',
 'ALIGN',
 'EYE',
 'HAIR',
 'SEX',
 'GSM',
 'ALIVE',
 'FIRST APPEARANCE',
 'Year']
```

In [26]:
```
# Доля (процент) пропусков
[(c, data[c].isnull().mean()) for c in cols_with_na]
```

Out[26]:
```
[('ID', 0.23021494870542256),
 ('ALIGN', 0.17171470444553005),
 ('EYE', 0.5964215925744992),
 ('HAIR', 0.26038104543234003),
 ('SEX', 0.05214948705422569 5),
 ('GSM', 0.9945041524181729),
 ('ALIVE', 0.0001831949193942 3546),
 ('FIRST APPEARANCE', 0.0497679531021 0064),
 ('Year', 0.0497679531021 0064)]
```

In [27]:
```
data.head()
```

Out[27]:

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEA
0	1678	Spider-Man (Peter Parker)	\/Spider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	
1	7139	Captain America (Steven Rogers)	\/Captain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	
2	64786	Wolverine (James \"Logan\" Howlett)	\/Wolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	Heterosexual	Living Characters	
3	1868	Iron Man (Anthony \"Tony\" Stark)	\/Iron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	Heterosexual	Living Characters	
4	2460	Thor (Thor Odinson)	\/Thor_(Thor_Odinson)	No Dual Identity	Good Characters	Blue Eyes	Blond Hair	Male Characters	Heterosexual	Living Characters	
...	...	...	...	...	...	...	...	...	...	...	...
370	674414	Phoenix's Shadow (Earth-616)	\/Phoenix%27s_Shadow_(Earth-616)	NaN	Neutral Characters	NaN	NaN	NaN	Heterosexual	Living Characters	

Дополнительное требование

[35]:
```
data.columns
```

Out[35]:
```
Index(['page_id', 'name', 'urlslug', 'ID', 'ALIGN', 'EYE', 'HAIR', 'SEX',
       'ALIVE', 'APPEARANCES', 'FIRST APPEARANCE', 'Year'],
      dtype='object')
```

[37]:
```
sns.pairplot(data)
```

Out[37]:
```
<seaborn.axisgrid.PairGrid at 0x16513af6d30>
```