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**

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

По курсу «Технологии машинного обучения»

**ИСПОЛНИТЕЛЬ:**

Коционова А. А.

Группа ИУ5-63

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2019 г.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю. Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2019 г.

Москва 2019

**Цель лабораторной работы.**

Изучить способы предварительной обработки данных для дальнейшего формирования моделей.

**Практическая часть.**

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**from** **sklearn.impute** **import** SimpleImputer, MissingIndicator

**from** **sklearn.preprocessing** **import** LabelEncoder, OneHotEncoder, MinMaxScaler,

StandardScaler, Normalizer

**import** **seaborn** **as** **sns**

**import** **matplotlib.pyplot** **as** **plt**

%**matplotlib** inline

sns.set(style="ticks")

**Характеристика выбранного датасета:**

**page\_id** | The unique identifier for that characters page within the wikia   
**name** | The name of the character   
**urlslug** | The unique url within the wikia that takes you to the character   
**ID** | The identity status of the character (Secret Identity, Public identity, [on marvel only: No Dual Identity])   
**ALIGN** | If the character is Good, Bad or Neutral   
**EYE** | Eye color of the character   
**HAIR** | Hair color of the character   
**SEX** | Sex of the character (e.g. Male, Female, etc.)   
**GSM** | If the character is a gender or sexual minority (e.g. Homosexual characters, bisexual characters)   
**ALIVE** | If the character is alive or deceased   
**APPEARANCES** | The number of appareances of the character in comic books (as of Sep. 2, 2014. Number will become increasingly out of date as time goes on.)   
**FIRST APPEARANCE** | The month and year of the character's first appearance in a comic book, if available   
**YEAR** | The year of the character's first appearance in a comic book, if available

In [251]:

dc = pd.read\_csv('C:/Users/kotsi/dc-wikia-data.csv')

**1.Обработка пропусков данных**

In [252]:

dc.head()

Out[252]:

|  | **page\_id** | **name** | **urlslug** | **ID** | **ALIGN** | **EYE** | **HAIR** | **SEX** | **GSM** | **ALIVE** | **APPEARANCES** | **FIRST APPEARANCE** | **YEAR** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1422 | Batman (Bruce Wayne) | \/wiki\/Batman\_(Bruce\_Wayne) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | 3093.0 | 1939, May | 1939.0 |
| **1** | 23387 | Superman (Clark Kent) | \/wiki\/Superman\_(Clark\_Kent) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | 2496.0 | 1986, October | 1986.0 |
| **2** | 1458 | Green Lantern (Hal Jordan) | \/wiki\/Green\_Lantern\_(Hal\_Jordan) | Secret Identity | Good Characters | Brown Eyes | Brown Hair | Male Characters | NaN | Living Characters | 1565.0 | 1959, October | 1959.0 |
| **3** | 1659 | James Gordon (New Earth) | \/wiki\/James\_Gordon\_(New\_Earth) | Public Identity | Good Characters | Brown Eyes | White Hair | Male Characters | NaN | Living Characters | 1316.0 | 1987, February | 1987.0 |
| **4** | 1576 | Richard Grayson (New Earth) | \/wiki\/Richard\_Grayson\_(New\_Earth) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | 1237.0 | 1940, April | 1940.0 |

In [253]:

dc.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 6896 entries, 0 to 6895

Data columns (total 13 columns):

page\_id 6896 non-null int64

name 6896 non-null object

urlslug 6896 non-null object

ID 4883 non-null object

ALIGN 6295 non-null object

EYE 3268 non-null object

HAIR 4622 non-null object

SEX 6771 non-null object

GSM 64 non-null object

ALIVE 6893 non-null object

APPEARANCES 6541 non-null float64

FIRST APPEARANCE 6827 non-null object

YEAR 6827 non-null float64

dtypes: float64(2), int64(1), object(10)

memory usage: 700.5+ KB

In [254]:

dc['FIRST APPEARANCE']=dc['FIRST APPEARANCE'].str[5:]

cat\_cols = [c **for** c **in** dc.columns **if** dc[c].dtype.name == 'object']

num\_cols=[c **for** c **in** dc.columns **if** dc[c].dtype.name != 'object']

In [255]:

dc[cat\_cols].describe()

Out[255]:

|  | **name** | **urlslug** | **ID** | **ALIGN** | **EYE** | **HAIR** | **SEX** | **GSM** | **ALIVE** | **FIRST APPEARANCE** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 6896 | 6896 | 4883 | 6295 | 3268 | 4622 | 6771 | 64 | 6893 | 6827 |
| **unique** | 6896 | 6896 | 3 | 4 | 17 | 17 | 4 | 2 | 2 | 14 |
| **top** | Ernest Widdle (New Earth) | \/wiki\/Rees-Van\_(New\_Earth) | Public Identity | Bad Characters | Blue Eyes | Black Hair | Male Characters | Homosexual Characters | Living Characters | August |
| **freq** | 1 | 1 | 2466 | 2895 | 1102 | 1574 | 4783 | 54 | 5200 | 634 |

In [256]:

cat\_dc=dc[cat\_cols]

cat\_dc.head()

Out[256]:

|  | **name** | **urlslug** | **ID** | **ALIGN** | **EYE** | **HAIR** | **SEX** | **GSM** | **ALIVE** | **FIRST APPEARANCE** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Batman (Bruce Wayne) | \/wiki\/Batman\_(Bruce\_Wayne) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | May |
| **1** | Superman (Clark Kent) | \/wiki\/Superman\_(Clark\_Kent) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | October |
| **2** | Green Lantern (Hal Jordan) | \/wiki\/Green\_Lantern\_(Hal\_Jordan) | Secret Identity | Good Characters | Brown Eyes | Brown Hair | Male Characters | NaN | Living Characters | October |
| **3** | James Gordon (New Earth) | \/wiki\/James\_Gordon\_(New\_Earth) | Public Identity | Good Characters | Brown Eyes | White Hair | Male Characters | NaN | Living Characters | February |
| **4** | Richard Grayson (New Earth) | \/wiki\/Richard\_Grayson\_(New\_Earth) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | April |

In [258]:

num\_dc=dc[num\_cols]

num\_dc.head()

Out[258]:

|  | **page\_id** | **APPEARANCES** | **YEAR** |
| --- | --- | --- | --- |
| **0** | 1422 | 3093.0 | 1939.0 |
| **1** | 23387 | 2496.0 | 1986.0 |
| **2** | 1458 | 1565.0 | 1959.0 |
| **3** | 1659 | 1316.0 | 1987.0 |
| **4** | 1576 | 1237.0 | 1940.0 |

In [259]:

dc.groupby('ID')['ID'].count()

Out[259]:

ID

Identity Unknown 9

Public Identity 2466

Secret Identity 2408

Name: ID, dtype: int64

In [260]:

dc.loc[dc["ID"].isnull()].head()

Out[260]:

|  | **page\_id** | **name** | **urlslug** | **ID** | **ALIGN** | **EYE** | **HAIR** | **SEX** | **GSM** | **ALIVE** | **APPEARANCES** | **FIRST APPEARANCE** | **YEAR** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **181** | 17885 | Samuel Morgan (New Earth) | \/wiki\/Samuel\_Morgan\_(New\_Earth) | NaN | Good Characters | NaN | Red Hair | Male Characters | NaN | Deceased Characters | 155.0 | March | 1937.0 |
| **203** | 113540 | Tubby Watts (New Earth) | \/wiki\/Tubby\_Watts\_(New\_Earth) | NaN | Good Characters | NaN | Red Hair | Male Characters | NaN | Living Characters | 137.0 | September | 1941.0 |
| **220** | 6063 | Victory (New Earth) | \/wiki\/Victory\_(New\_Earth) | NaN | Good Characters | Black Eyes | White Hair | Male Characters | NaN | Living Characters | 130.0 | September | 1941.0 |
| **251** | 1841 | Dolphin (New Earth) | \/wiki\/Dolphin\_(New\_Earth) | NaN | Good Characters | Blue Eyes | Silver Hair | Female Characters | NaN | Deceased Characters | 115.0 | December | 1968.0 |
| **277** | 341732 | Sugar Plumm (New Earth) | \/wiki\/Sugar\_Plumm\_(New\_Earth) | NaN | Good Characters | NaN | Blond Hair | Female Characters | NaN | Living Characters | 105.0 | May | 1956.0 |

**Личность всех этих персонажей можно охарактеризовать как неизвестную**

In [261]:

dc['ID']=dc["ID"].fillna('Identity Unknown')

In [262]:

dc.groupby('ALIGN')['ALIGN'].count()

Out[262]:

ALIGN

Bad Characters 2895

Good Characters 2832

Neutral Characters 565

Reformed Criminals 3

Name: ALIGN, dtype: int64

In [263]:

dc['ALIGN']=dc["ALIGN"].fillna('Neutral Characters')

In [264]:

dc.groupby('ALIGN')['ALIGN'].count()

Out[264]:

ALIGN

Bad Characters 2895

Good Characters 2832

Neutral Characters 1166

Reformed Criminals 3

Name: ALIGN, dtype: int64

**Следующие признаки либо маловажны, либо заполнены практически целиком, так что их мы заполним наиболее частыми**

In [265]:

data\_describe = dc.describe(include=[object])

**for** c **in** ['EYE','HAIR','SEX','ALIVE','FIRST APPEARANCE']:

dc[c] = dc[c].fillna(data\_describe[c]['top'])

dc.head()

Out[265]:

|  | **page\_id** | **name** | **urlslug** | **ID** | **ALIGN** | **EYE** | **HAIR** | **SEX** | **GSM** | **ALIVE** | **APPEARANCES** | **FIRST APPEARANCE** | **YEAR** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1422 | Batman (Bruce Wayne) | \/wiki\/Batman\_(Bruce\_Wayne) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | 3093.0 | May | 1939.0 |
| **1** | 23387 | Superman (Clark Kent) | \/wiki\/Superman\_(Clark\_Kent) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | 2496.0 | October | 1986.0 |
| **2** | 1458 | Green Lantern (Hal Jordan) | \/wiki\/Green\_Lantern\_(Hal\_Jordan) | Secret Identity | Good Characters | Brown Eyes | Brown Hair | Male Characters | NaN | Living Characters | 1565.0 | October | 1959.0 |
| **3** | 1659 | James Gordon (New Earth) | \/wiki\/James\_Gordon\_(New\_Earth) | Public Identity | Good Characters | Brown Eyes | White Hair | Male Characters | NaN | Living Characters | 1316.0 | February | 1987.0 |
| **4** | 1576 | Richard Grayson (New Earth) | \/wiki\/Richard\_Grayson\_(New\_Earth) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | NaN | Living Characters | 1237.0 | April | 1940.0 |

**Введем гетеросексуальную ориентацию и предположим, что все остальные супергерои действительно таковы**

In [266]:

dc['GSM']=dc["GSM"].fillna('Heterosexual Characters')

**Посмотрим, все ли заполнили**

In [267]:

dc.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 6896 entries, 0 to 6895

Data columns (total 13 columns):

page\_id 6896 non-null int64

name 6896 non-null object

urlslug 6896 non-null object

ID 6896 non-null object

ALIGN 6896 non-null object

EYE 6896 non-null object

HAIR 6896 non-null object

SEX 6896 non-null object

GSM 6896 non-null object

ALIVE 6896 non-null object

APPEARANCES 6541 non-null float64

FIRST APPEARANCE 6896 non-null object

YEAR 6827 non-null float64

dtypes: float64(2), int64(1), object(10)

memory usage: 700.5+ KB

**Заполним числовые признаки**

In [268]:

dc[num\_cols] = dc[num\_cols].fillna(dc[num\_cols].median(axis=0), axis=0)

**Проеведем еще небольшое преобразование данных, отсечем год в столбце "Первое появление" и преобразуем некоторые типы**

In [270]:

dc[['YEAR','APPEARANCES']]=dc[['YEAR','APPEARANCES']].astype(int)

dc.head()

Out[270]:

|  | **page\_id** | **name** | **urlslug** | **ID** | **ALIGN** | **EYE** | **HAIR** | **SEX** | **GSM** | **ALIVE** | **APPEARANCES** | **FIRST APPEARANCE** | **YEAR** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1422 | Batman (Bruce Wayne) | \/wiki\/Batman\_(Bruce\_Wayne) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | Heterosexual Characters | Living Characters | 3093 | May | 1939 |
| **1** | 23387 | Superman (Clark Kent) | \/wiki\/Superman\_(Clark\_Kent) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | Heterosexual Characters | Living Characters | 2496 | October | 1986 |
| **2** | 1458 | Green Lantern (Hal Jordan) | \/wiki\/Green\_Lantern\_(Hal\_Jordan) | Secret Identity | Good Characters | Brown Eyes | Brown Hair | Male Characters | Heterosexual Characters | Living Characters | 1565 | October | 1959 |
| **3** | 1659 | James Gordon (New Earth) | \/wiki\/James\_Gordon\_(New\_Earth) | Public Identity | Good Characters | Brown Eyes | White Hair | Male Characters | Heterosexual Characters | Living Characters | 1316 | February | 1987 |
| **4** | 1576 | Richard Grayson (New Earth) | \/wiki\/Richard\_Grayson\_(New\_Earth) | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | Heterosexual Characters | Living Characters | 1237 | April | 1940 |

In [271]:

dc.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 6896 entries, 0 to 6895

Data columns (total 13 columns):

page\_id 6896 non-null int64

name 6896 non-null object

urlslug 6896 non-null object

ID 6896 non-null object

ALIGN 6896 non-null object

EYE 6896 non-null object

HAIR 6896 non-null object

SEX 6896 non-null object

GSM 6896 non-null object

ALIVE 6896 non-null object

APPEARANCES 6896 non-null int32

FIRST APPEARANCE 6896 non-null object

YEAR 6896 non-null int32

dtypes: int32(2), int64(1), object(10)

memory usage: 646.6+ KB

**Кодирование категориальных признаков**

In [272]:

binary\_headers=[]

non\_binary\_headers=[]

[binary\_headers.append(c) **for** c **in** cat\_cols **if** dc[c].nunique()==2 ]

[non\_binary\_headers.append(c) **for** c **in** cat\_cols **if** dc[c].nunique()!=2 **and** c **not** **in** ('name','urlslug') ]

binary\_columns=dc[binary\_headers]

non\_binary\_columns=dc[non\_binary\_headers]

In [273]:

le = LabelEncoder()

cat\_enc\_le = le.fit\_transform(non\_binary\_columns['ALIGN'])

*#находит все уникальные значения и строит таблицу для соответствия каждой категории*

*#некоторому числу, затем преобразует значения в числа*

non\_binary\_columns['ALIGN'].unique()

Out[273]:

array(['Good Characters', 'Bad Characters', 'Neutral Characters',

'Reformed Criminals'], dtype=object)

In [274]:

*#le.inverse\_transform([0, 1, 2, 3])-обратная трансформация*

np.unique(cat\_enc\_le)

Out[274]:

array([0, 1, 2, 3])

**результат**

In [275]:

*### Так как подход так себе, обратимся к OneHot Encoding-у*

non\_binary\_columns.head()

Out[275]:

|  | **ID** | **ALIGN** | **EYE** | **HAIR** | **SEX** | **GSM** | **FIRST APPEARANCE** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | Heterosexual Characters | May |
| **1** | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | Heterosexual Characters | October |
| **2** | Secret Identity | Good Characters | Brown Eyes | Brown Hair | Male Characters | Heterosexual Characters | October |
| **3** | Public Identity | Good Characters | Brown Eyes | White Hair | Male Characters | Heterosexual Characters | February |
| **4** | Secret Identity | Good Characters | Blue Eyes | Black Hair | Male Characters | Heterosexual Characters | April |

In [276]:

dc\_cat = pd.get\_dummies(dc[non\_binary\_headers])

dc\_cat.head()

Out[276]:

|  | **ID\_Identity Unknown** | **ID\_Public Identity** | **ID\_Secret Identity** | **ALIGN\_Bad Characters** | **ALIGN\_Good Characters** | **ALIGN\_Neutral Characters** | **ALIGN\_Reformed Criminals** | **EYE\_Amber Eyes** | **EYE\_Auburn Hair** | **EYE\_Black Eyes** | **...** | **FIRST APPEARANCE\_ February** | **FIRST APPEARANCE\_ Holiday** | **FIRST APPEARANCE\_ January** | **FIRST APPEARANCE\_ July** | **FIRST APPEARANCE\_ June** | **FIRST APPEARANCE\_ March** | **FIRST APPEARANCE\_ May** | **FIRST APPEARANCE\_ November** | **FIRST APPEARANCE\_ October** | **FIRST APPEARANCE\_ September** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **1** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **2** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **3** | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

5 rows × 62 columns

In [277]:

dc.info()

num\_cols

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 6896 entries, 0 to 6895

Data columns (total 13 columns):

page\_id 6896 non-null int64

name 6896 non-null object

urlslug 6896 non-null object

ID 6896 non-null object

ALIGN 6896 non-null object

EYE 6896 non-null object

HAIR 6896 non-null object

SEX 6896 non-null object

GSM 6896 non-null object

ALIVE 6896 non-null object

APPEARANCES 6896 non-null int32

FIRST APPEARANCE 6896 non-null object

YEAR 6896 non-null int32

dtypes: int32(2), int64(1), object(10)

memory usage: 646.6+ KB

Out[277]:

['page\_id', 'APPEARANCES', 'YEAR']

**Масштабирование данных**

In [278]:

sc1 = MinMaxScaler()

dc1=[sc1.fit\_transform(dc[[c]]) **for** c **in** ('APPEARANCES', 'YEAR')]

C:\Users\kotsi\Anaconda37\lib\site-packages\sklearn\preprocessing\data.py:323: DataConversionWarning: Data with input dtype int32 were all converted to float64 by MinMaxScaler.

return self.partial\_fit(X, y)

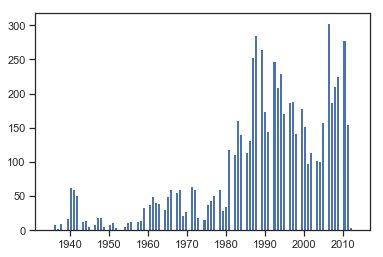
C:\Users\kotsi\Anaconda37\lib\site-packages\sklearn\preprocessing\data.py:323: DataConversionWarning: Data with input dtype int32 were all converted to float64 by MinMaxScaler.

return self.partial\_fit(X, y)

In [279]:

plt.hist(dc['YEAR'], 100)

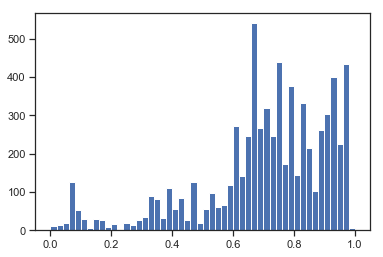
plt.show()



In [280]:

plt.hist(dc1[1], 50)

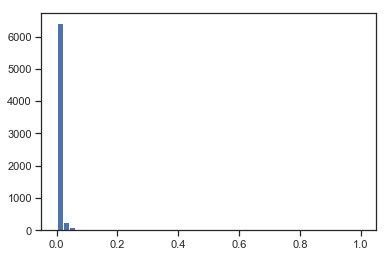
plt.show()



In [281]:

plt.hist(dc1[0], 50)

plt.show()



In [282]:

sc2 = StandardScaler()

sc2\_data = sc2.fit\_transform(dc[['YEAR']])

C:\Users\kotsi\Anaconda37\lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning: Data with input dtype int32 were all converted to float64 by StandardScaler.

return self.partial\_fit(X, y)

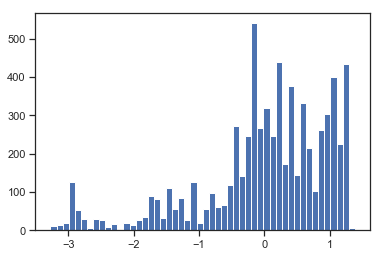
C:\Users\kotsi\Anaconda37\lib\site-packages\sklearn\base.py:462: DataConversionWarning: Data with input dtype int32 were all converted to float64 by StandardScaler.

return self.fit(X, \*\*fit\_params).transform(X)

In [283]:

plt.hist(sc2\_data, 50)

plt.show()



In [284]:

sc3 = Normalizer()

dc2= sc3.fit\_transform(dc[['APPEARANCES', 'YEAR']])

In [285]:

dc2=pd.DataFrame(dc2)

In [286]:

dc2.columns=['APPEARANCES', 'YEAR']

In [287]:

dc2.head()

Out[287]:

|  | **APPEARANCES** | **YEAR** |
| --- | --- | --- |
| **0** | 0.847274 | 0.531156 |
| **1** | 0.782518 | 0.622628 |
| **2** | 0.624160 | 0.781297 |
| **3** | 0.552180 | 0.833725 |
| **4** | 0.537635 | 0.843178 |

In [288]:

dc2.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 6896 entries, 0 to 6895

Data columns (total 2 columns):

APPEARANCES 6896 non-null float64

YEAR 6896 non-null float64

dtypes: float64(2)

memory usage: 107.8 KB

In [289]:

DC\_BIG = pd.concat([dc\_cat,dc2], axis=1)

In [290]:

DC\_BIG.head()

Out[290]:

|  | **ID\_Identity Unknown** | **ID\_Public Identity** | **ID\_Secret Identity** | **ALIGN\_Bad Characters** | **ALIGN\_Good Characters** | **ALIGN\_Neutral Characters** | **ALIGN\_Reformed Criminals** | **EYE\_Amber Eyes** | **EYE\_Auburn Hair** | **EYE\_Black Eyes** | **...** | **FIRST APPEARANCE\_ January** | **FIRST APPEARANCE\_ July** | **FIRST APPEARANCE\_ June** | **FIRST APPEARANCE\_ March** | **FIRST APPEARANCE\_ May** | **FIRST APPEARANCE\_ November** | **FIRST APPEARANCE\_ October** | **FIRST APPEARANCE\_ September** | **APPEARANCES** | **YEAR** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0.847274 | 0.531156 |
| **1** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0.782518 | 0.622628 |
| **2** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0.624160 | 0.781297 |
| **3** | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.552180 | 0.833725 |
| **4** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.537635 | 0.843178 |

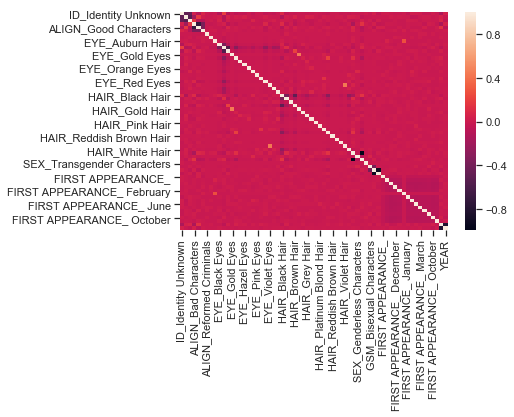
5 rows × 64 columns

In [291]:

sns.heatmap(DC\_BIG.corr())

Out[291]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x1bb81ace390>



In [292]:

print(DC\_BIG.corr()['APPEARANCES'].abs().sort\_values(ascending=**False**).head(10))

APPEARANCES 1.000000

YEAR 0.818192

ALIGN\_Good Characters 0.141453

ALIGN\_Bad Characters 0.140337

ID\_Identity Unknown 0.137600

ID\_Secret Identity 0.080220

EYE\_Blue Eyes 0.062221

ID\_Public Identity 0.050905

EYE\_Green Eyes 0.050818

HAIR\_Black Hair 0.050293

Name: APPEARANCES, dtype: float64