data-presentation

May 18, 2021

1 Présentation des données

Dans ce notebook, nous allons commencer par prendre en main les données fournit.

[ ]:
```
#!pip3 install missingno
```

[1]:
```
import pandas as pd
import missingno as msno
```

[2]:
```
file_name = "../data/claimKG.csv"

# Lecture du fichier
kg = pd.read_csv(file_name)
```

[3]:
```
origin = kg.copy()

nb_assertion = len(kg)
print("Le nombre d'assertions : ", nb_assertion)
```

```
Le nombre d'assertions :  39218
```

1.1 Données manquantes

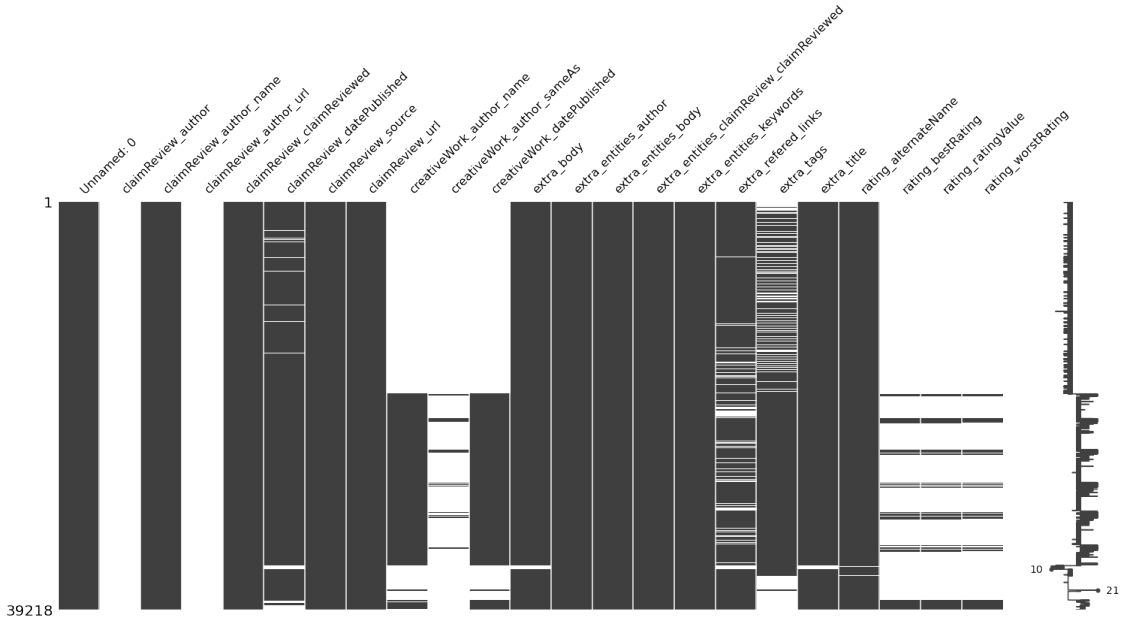
On voit bien sur le schéma ci-dessous les données qui manquent:

- ClaimReview_author
- ClaimReview_author_url
- creativeWork_author_name
- creativeWork_author_sameAs
- creativeWork_datePublished
- rating_bestRating
- rating_ratingValue
- rating_worstRating

[4]:
```
print("Les données manquantes: ")
msno.matrix(origin)
```

```
Les données manquantes:
```

[4]: <AxesSubplot:>



On voit ici que certaines colonnes sont complétement ou majoritairement manquantes. Il faut donc les supprimer, car ils ne nous servirons pas pour entraîner notre modèle.

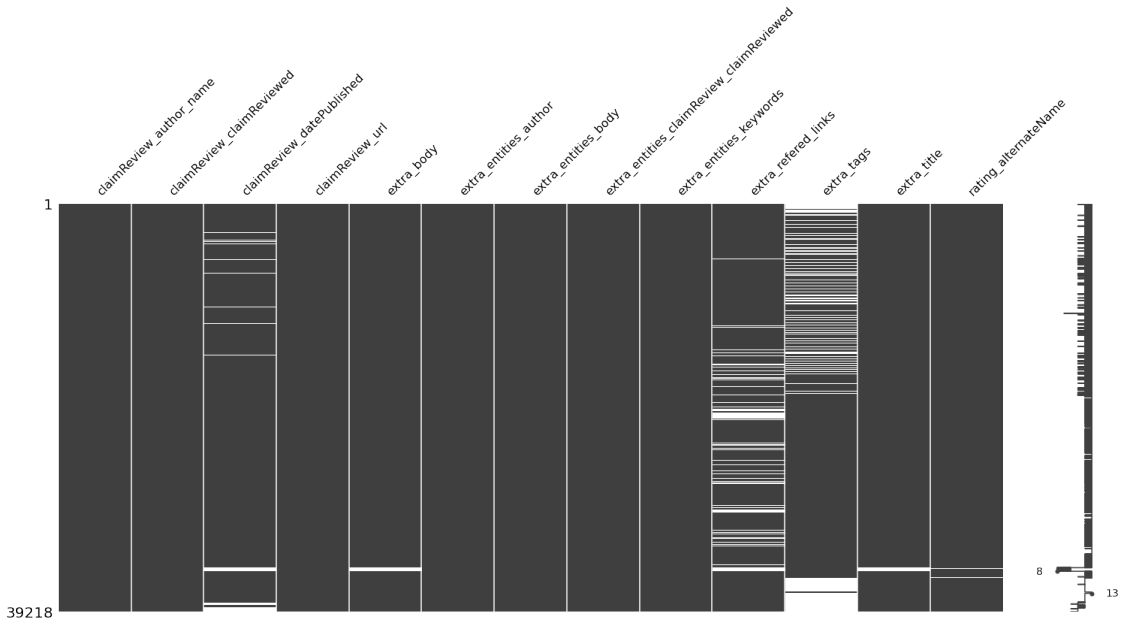
[5]:
```
manq = origin.isnull().sum()
col_manq = ['Unnamed: 0','claimReview_source']
for k, v in manq.items():
    if v > nb_assertion/2 :
        col_manq.append(k)

print("Les colonnes qui seront supprimés: ")
for col in col_manq:
    print("->",col)
```

```
Les colonnes qui seront supprimés:
-> Unnamed: 0
-> claimReview_source
-> claimReview_author
-> claimReview_author_url
-> creativeWork_author_name
-> creativeWork_author_sameAs
-> creativeWork_datePublished
-> rating_bestRating
-> rating_ratingValue
-> rating_worstRating
```

Une fois enlever, on obtient le schéma suivant.

[6]:
```
kg.drop(columns=col_manq, inplace=True)
msno.matrix(kg)
```

[6]: <AxesSubplot:>



1.2 Doublons

[7]:
```
duplicateRows = kg[kg.duplicated()]

print("Nombre de lignes redondantes :", len(duplicateRows))
```

```
Nombre de lignes redondantes : 5749
```

On voit ici qu'il y a un certain nombre de lignes redondantes. Il faut alors les supprimer parcequ'elles risquent d'apporter une certaine imprécision.

[8]:
```
d = [i for i, v in kg.duplicated().items() if v]
kg.drop(d,inplace=True)
```

1.3 Autres erreurs dans les données

En parcourant les données, on a remarqué quelques incohérences dans les données qu'on préfère éliminer dès maintenant.

Notamment, certaines lignes contiennent comme claim `true` ou `false`

[9]:
```
mask = kg['claimReview_claimReviewed'].isin(["false","true"])
display(kg[mask]['claimReview_claimReviewed'])
```

```
rm = set(kg[mask].index.values)
```

```
8212     false
11863    false
12054     true
Name: claimReview_claimReviewed, dtype: object
```

D'autres lignes contiennent rien comme claim. Sans claim, on ne pourra rien faire.

[10]:
```
mask = kg['claimReview_claimReviewed'].isnull()
display(kg[mask]['claimReview_claimReviewed'])
rm = rm.union(set(kg[mask].index.values))
```

```
38909    NaN
39178    NaN
Name: claimReview_claimReviewed, dtype: object
```

[11]:
```
kg.drop(rm,inplace=True)
print("Nombre d'assertion:",len(kg))
```

```
Nombre d'assertion: 33464
```

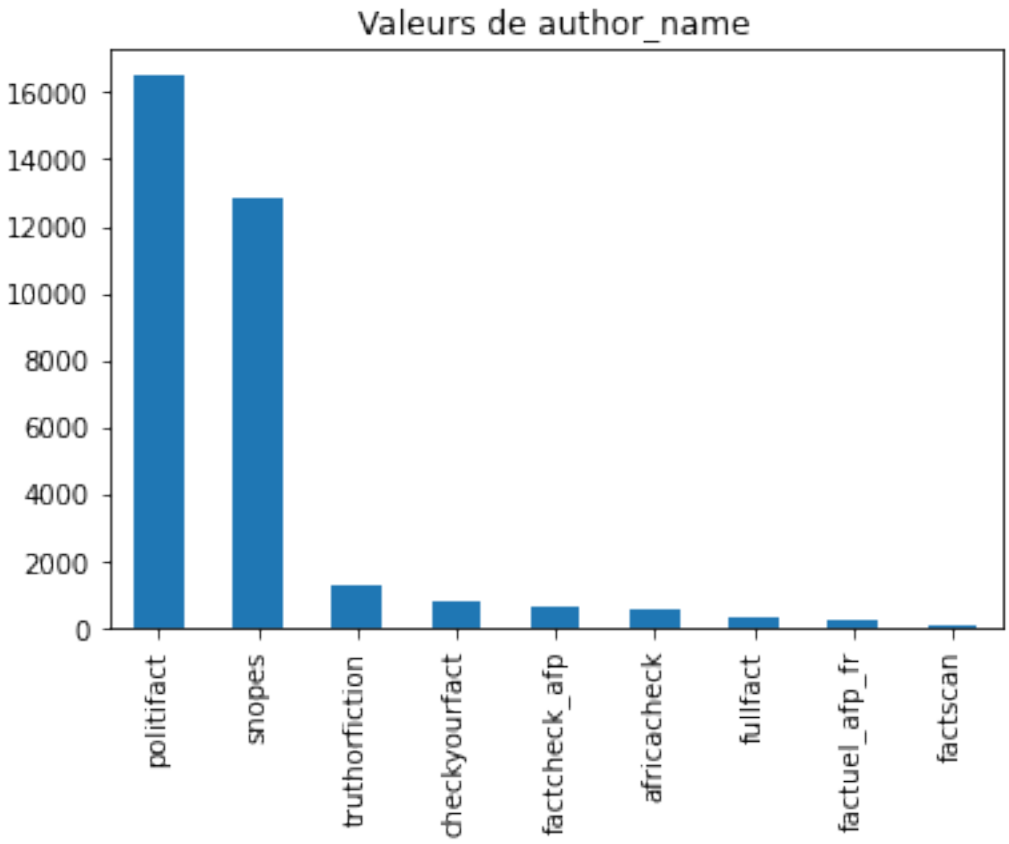
1.4 Visualisation des données

1.4.1 claimReview_author_name

Cet attribut contient le nom de la source de l'assertion.

[12]:
```
kg['claimReview_author_name'].value_counts().plot(kind='bar',title="Valeurs de␣
 ↪author_name")
```

[12]:
```
<AxesSubplot:title={'center':'Valeurs de author_name'}>
```



On voit sur ce graphe que la majorité des assertions viennent de `politifact` et `snopes`.

1.4.2 claimReview_claimReviewed

Cet attribut contient l'assertion en question.

[13]:
```
s = kg[['claimReview_claimReviewed']].sample(5)
display(s)
```

```
                                claimReview_claimReviewed
19691  ""$16 trillion (the national debt) in $1 bills…
38038  Are 80% Of Migrant Women Raped On Their Journe…
18255  Sororities are outlawed on certain campuses be…
38550  ‘Betrayed and murdered the Kurdish people’ ban…
12262  Soon after winning the largest-ever Megabucks …
```

NB: Certaine assertions venant de `factuel_afp_fr` et `factcheck_afp` sont en français. Il faut surtout pas les mélanger avec les autres.

Note à plus tard: Trouvez un moyen de séparer proprement les assertions selon leurs langues.

[14]:
```
fr_author = ['factcheck_afp', 'factuel_afp_fr']
fr = kg[kg['claimReview_author_name'].
 ↪isin(fr_author)][['claimReview_claimReviewed']]
print("Nombre d'assertion de",fr_author," : ",len(fr))
display(fr.sample(5))
```

```
Nombre d'assertion de ['factcheck_afp', 'factuel_afp_fr']  :  942
```

	claimReview_claimReviewed
38762	Genuine report about the Philippine vice presi…
38394	Un ""enregistrement secret"" de Macky Sall cir…
39011	CCTV footage showing bomb attack in Kashmir, P…
38568	Photos show Communist party of India (Marxits)…
38297	Robert Mugabe a un cercueil informatisé qui va…

1.4.3 claimReview_datePublished

Cet attribut contient la date de publication de l'assertion.

[15]:
```
display(kg[['claimReview_datePublished']].sample(3))
print("Nombre de date manquantes:", len(kg[kg['claimReview_datePublished'].
 ↪isnull()]))
```

	claimReview_datePublished
14512	2011-08-11
1282	2019-01-04
33096	2019-02-18

```
Nombre de date manquantes: 1301
```

1.4.4 claimReview_url

Lien vers l'article qui parle de cette assertion

[16]:
```
display(kg[['claimReview_url']].sample(3))
print("Nombre de liens manquants:",len(kg[kg['claimReview_url'].isnull()]))
```

	claimReview_url
7174	https://www.snopes.com/fact-check/hillary-sand…
20240	http://www.politifact.com/truth-o-meter/statem…
12608	https://www.snopes.com/fact-check/katelyn-mari…

```
Nombre de liens manquants: 0
```

1.5 Premier traitement sur les données

On a vu que certaines données ne seront pas utilisable pour l'entraînement du modèle. Voici donc une fonction qui prend en paramètre le dataframe et supprimes toutes les colonnes et lignes à supprimer avant de commencer le prétraitement.

[17]:
```python
def clean_claimKG(df, verbose=False, inplace=False):
    if verbose:
        print("Taille du dataframe:",df.shape)
    nb_assertion = len(df)
    manq = df.isnull().sum()
    col_manq = ['Unnamed: 0','claimReview_source']
    for k, v in manq.items():
        if v > nb_assertion/2 :
            col_manq.append(k)
    if verbose:
        print("Suppression des columns suivant:",col_manq)
    df = df.drop(columns=col_manq,inplace=inplace)

    d = [i for i, v in df.duplicated().items() if v]
    if verbose:
        print("Suppression de",len(d),"lignes en doubles.")
    df = df.drop(d,inplace=inplace)

    mask = df['claimReview_claimReviewed'].isin(["false","true"])
    rm = set(df[mask].index.values)
    mask = df['claimReview_claimReviewed'].isnull()
    rm = rm.union(set(df[mask].index.values))

    if verbose:
        print("Suppression de",len(rm),"lignes.")
    df = df.drop(rm, inplace=inplace)

    if verbose:
        print("Taille finale:",df.shape)
    return df

kg_test = clean_claimKG(origin,verbose=True)
```

```
Taille du dataframe: (39218, 23)
Suppression des columns suivant: ['Unnamed: 0', 'claimReview_source',
'claimReview_author', 'claimReview_author_url', 'creativeWork_author_name',
'creativeWork_author_sameAs', 'creativeWork_datePublished', 'rating_bestRating',
'rating_ratingValue', 'rating_worstRating']
Suppression de 5749 lignes en doubles.
Suppression de 5 lignes.
Taille finale: (33464, 13)
```

Ce code est disponible sous clean.py pour réutilisation plus tard.

La prochaine étape est le prétraitement (tokenisation, lemmatisation, ...). On se retrouve sur le notebook pretraitement.ipynb.