Classifiez automatiquement des biens de consommation

Parcours Data Scientist - *OPENCLASSROOMS*

Plan de la présentation

1. Problématique et données
    a. Rappel de la problématique
    b. Présentation du jeu de données
    c. Récupération des données manquantes
2. Etude des données textuelles
    a. Exploration du corpus des descriptions
    b. Classification non supervisée
    c. Classification supervisée
3. Etude des données visuelles
    a. Prétraitement des images
    b. Extraction des features
    c. Classification des images
    d. Utilisation d'un réseau de neurones
4. Conclusion

Problématique et données

Rappel de la problématique

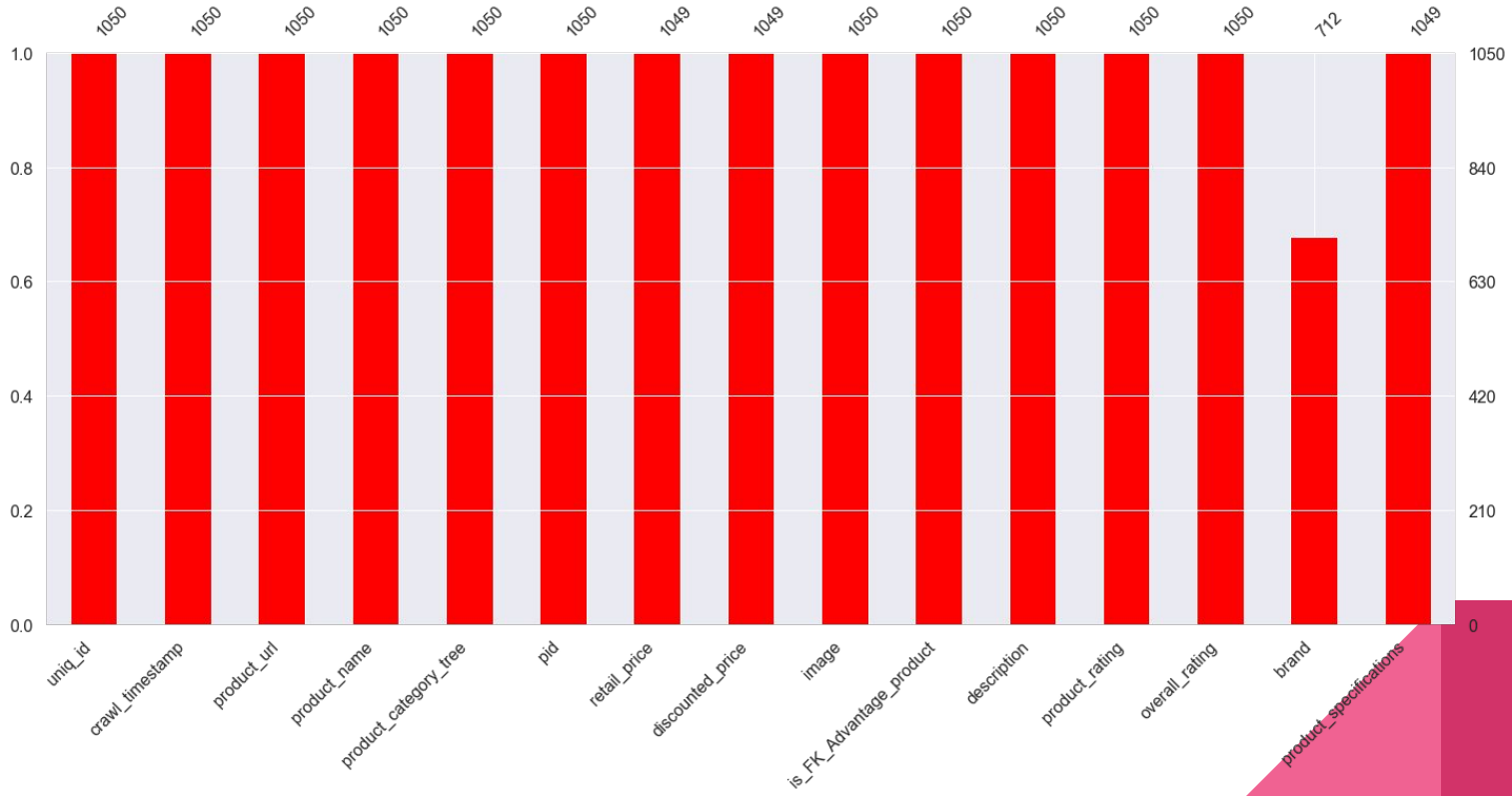
Rappel de la problématique

- “Place de marché” : classification automatique des biens par image et par description
- Etude de la faisabilité de ce moteur de classification
- Comment récupérer plus de données
- Analyser le jeu de données

Problématique et données

Présentation du jeu de données

Présentation du jeu de données



Présentation du jeu de données

… et des images de produits

Présentation du jeu de données

category depth 0	
Computers	150
Kitchen & Dining	150
Beauty and Personal Care	150
Watches	150
Home Decor & Festive Needs	150
Baby Care	150
Home Furnishing	150

Problématique et données

Récupération des données manquantes

Récupération des données manquantes

- Utilisation de pyaws : https://pypi.org/project/pyaws/
- Faire un appel http pour récupérer le xml :

http://webservices.amazon.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate ID]&
Operation=ItemSearch&
VariationPage=1&
Sort=salesrank&
Keywords=[Product description / name]&
SearchIndex=[Category du produit]&
Signature=[Tracker Id de la requête]

Etude des données textuelles

Exploration du corpus des descriptions

Exploration du corpus des descriptions

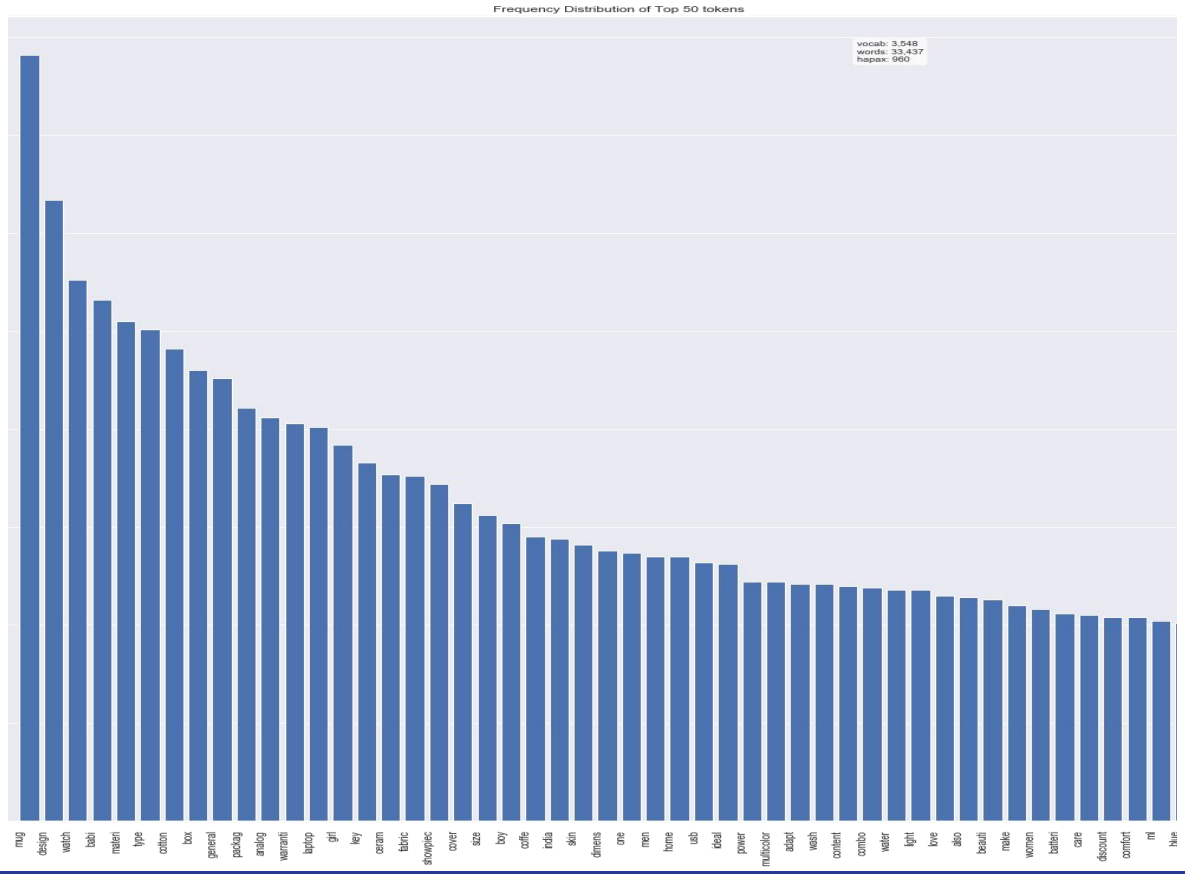
- Utilisation de stop words anglais + ponctuation + mots très utilisés dans la description
- Utilisation d’un stemmer (SnowBallStemmer)
- On s’est limité aux mots (avec une regex)

Exploration du corpus des descriptions

n	3
analog	2
men	2
timewel	2
watch	2
discount	1
india	1

- On applique un count vectorizer aux descriptions

Exploration du corpus des descriptions

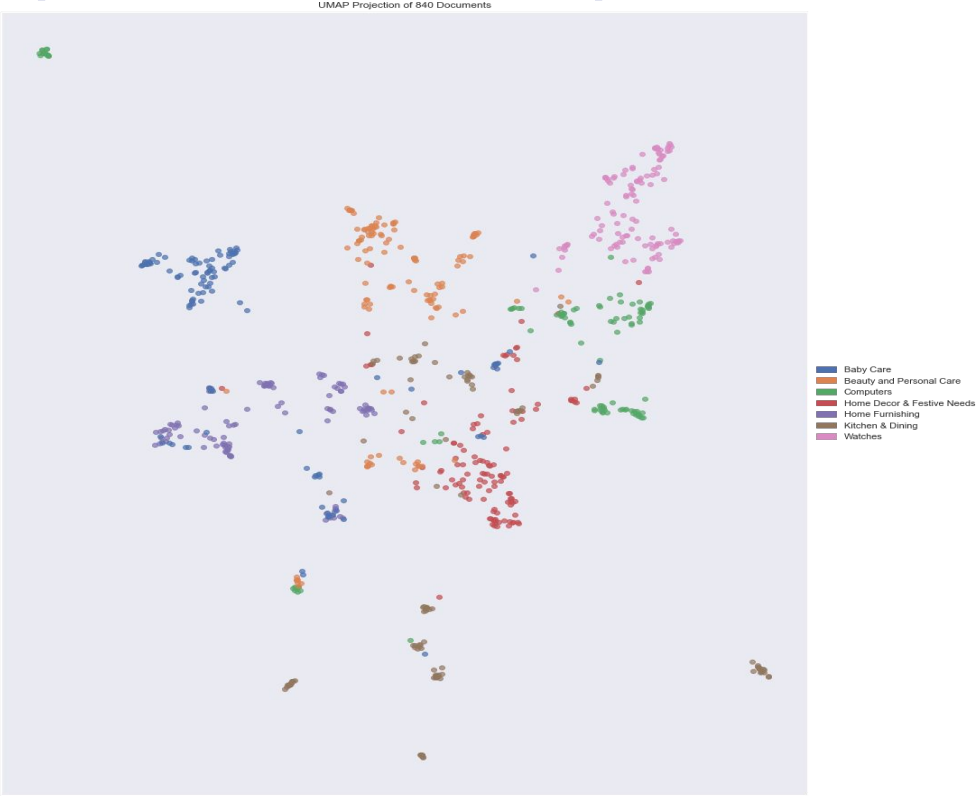


Exploration du corpus des descriptions

n	0.621586
timewel	0.540787
men	0.337555
analog	0.295508
watch	0.281823
discount	0.147296
india	0.137084

- En appliquant une transformation TF-IDF

Exploration du corpus des descriptions



Etude des données textuelles

Classification non supervisée

Classification non supervisée

- Utilisation d’une NMF:

Topic #0: mug ceram coffe ml prithish tea one love design get

Topic #1: watch analog men india women discount dial strap sonata maxima

Topic #2: babi cotton girl fabric dress ideal boy general content neck

Topic #3: craft rockmantra come fresh ensur exclus creation year design yet

Topic #4: singl abstract blanket doubl multicolor comfort quilt cover cushion floral

Topic #5: combo laptop usb warranti batteri skin led power cell light

Topic #6: showpiec towel kadhai sticker bath brass n router decor handicraft

Classification non supervisée

- Utilisation d’une LDA:

Topic #0: curtain tenda eyelet polyest door n combo hair skin kadhai

Topic #1: kadhai timewel pp lamp kalash runner hub link tabl combo

Topic #2: mug design cover showpiec multicolor warranti ceram box home cushion

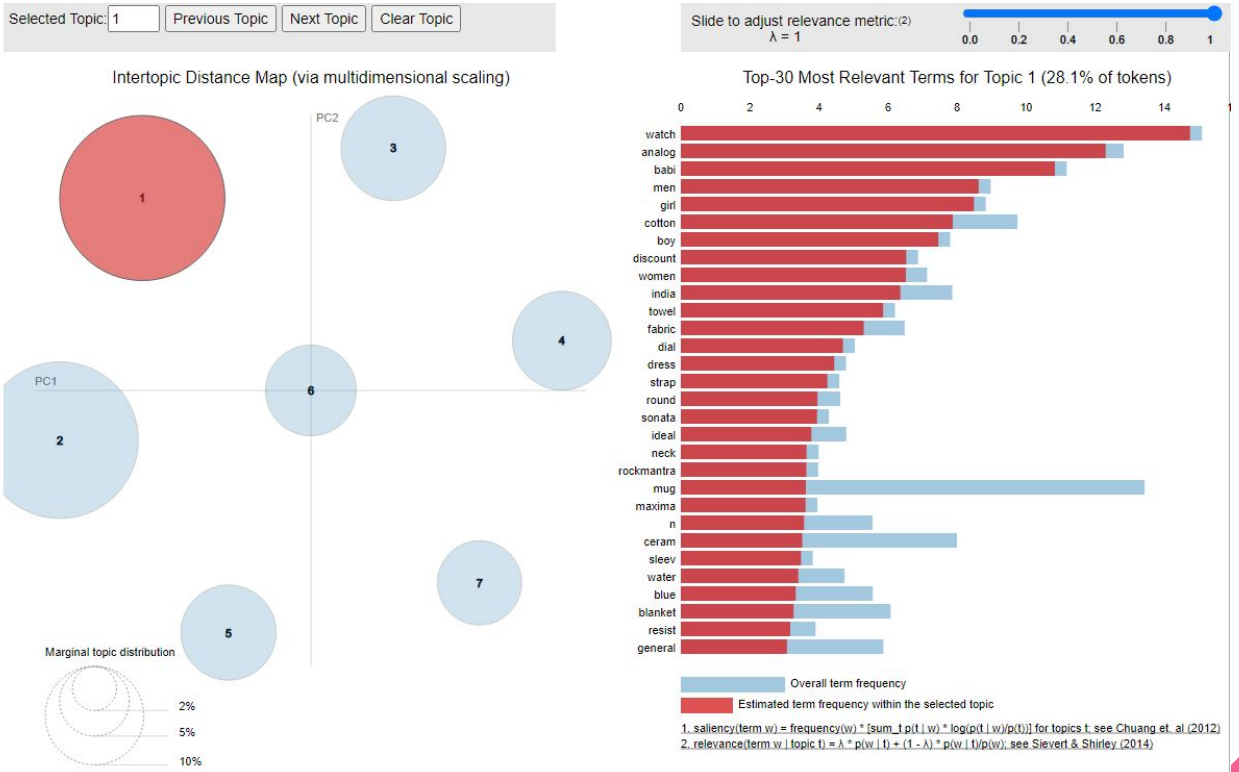
Topic #3: combo quilt showpiec comfort singl floral jewelleri handicraft playboy abstract

Topic #4: usb led light bottl bulb power flexibl nutcas portabl showpiec

Topic #5: laptop coffe mug cell pc hp batteri dv skin pavilion

Topic #6: watch analog babi men girl cotton boy discount women india

Classification non supervisée



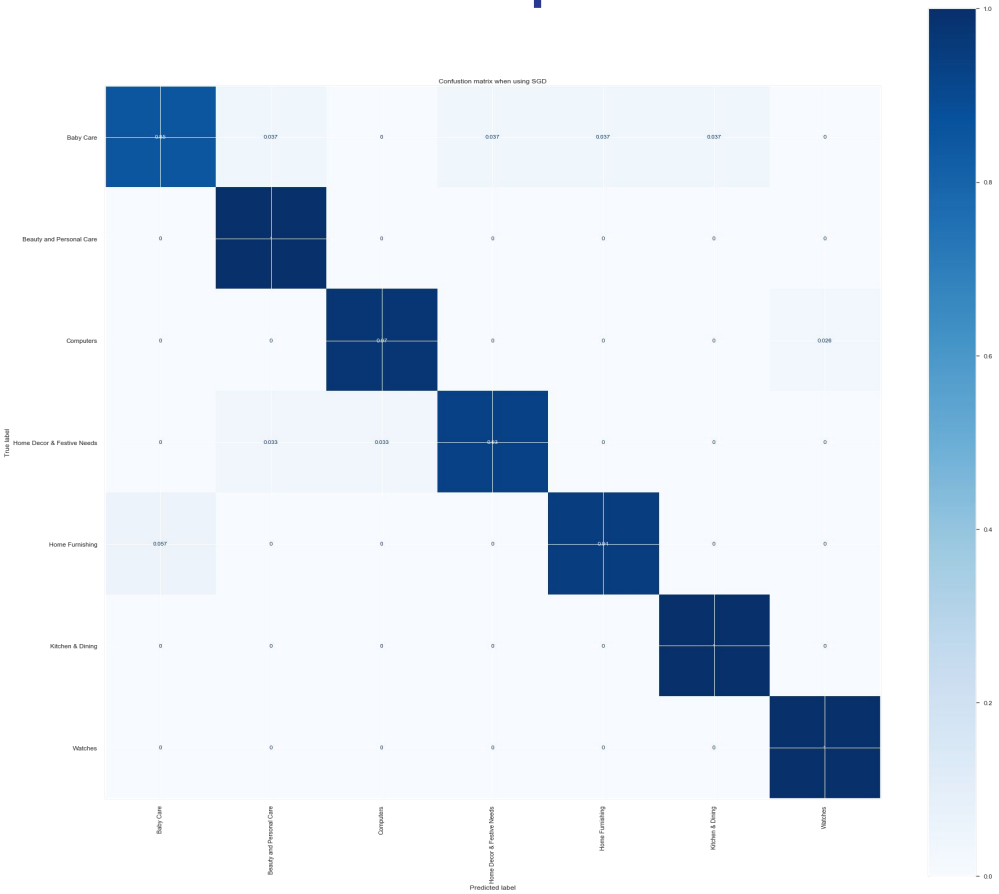
Etude des données textuelles

Classification supervisée

Classification supervisée

- Utilisation du SGD pour un testing score : 0.95 (cross val score 0.94) :
- Les paramètres optimaux en utilisant un pipeline :
  - tfidf__norm: 'l1'
  - tfidf__use_idf: False
  - vect__encoding: 'utf-8'
  - vect__lowercase: True
  - vect__max_df: 0.7
  - vect__min_df: 1
  - vect__ngram_range: (1, 1)

Classification supervisée

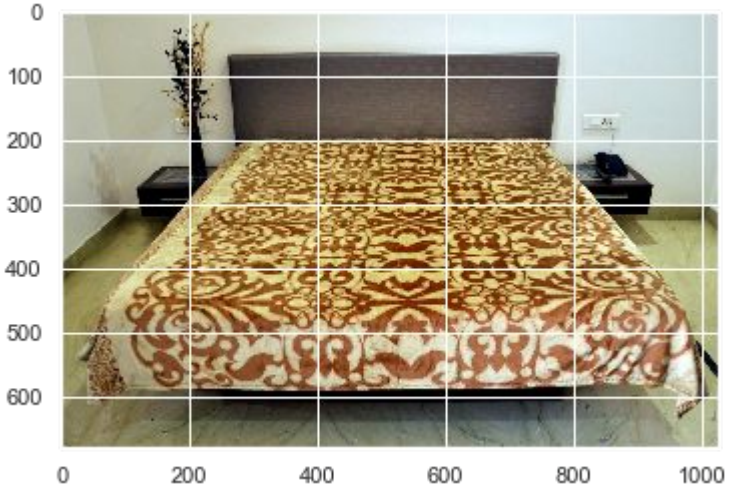


Etude des données visuelles

Prétraitement des images

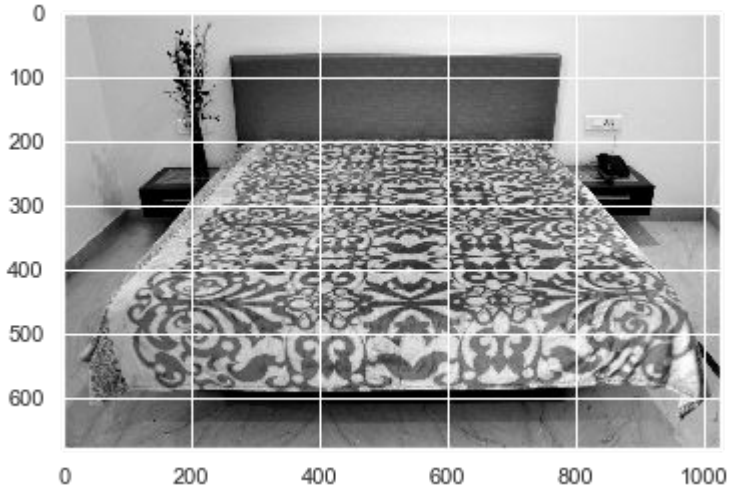
Prétraitement des images

- Image brute:



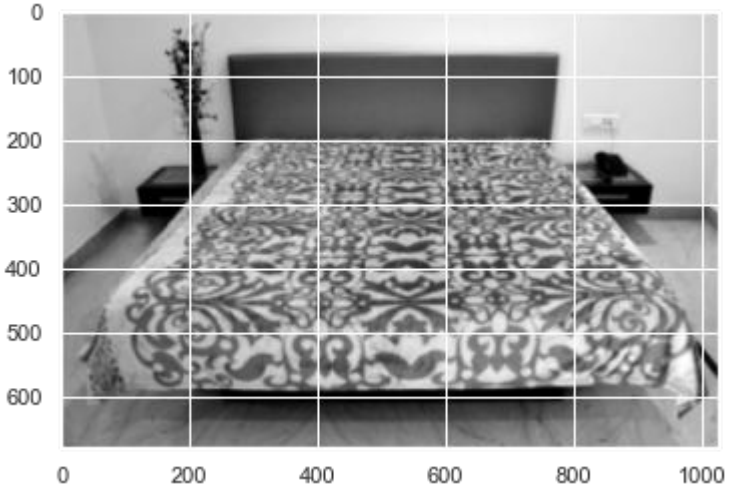
Prétraitement des images

- Passage en gris :



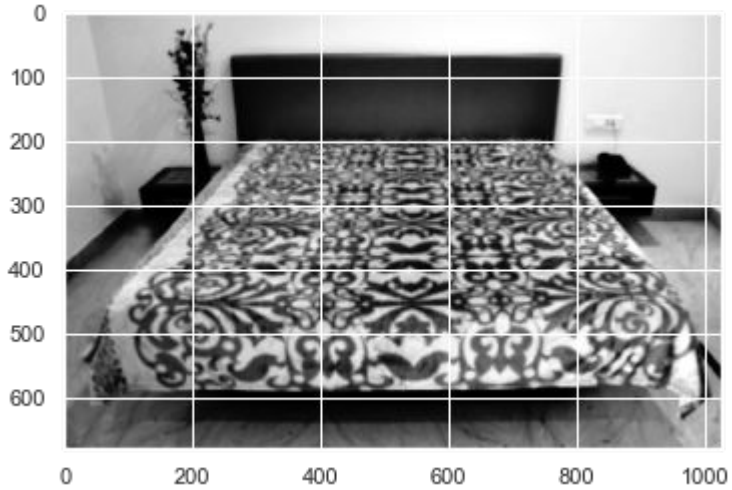
Prétraitement des images

- Floutage de l’image :



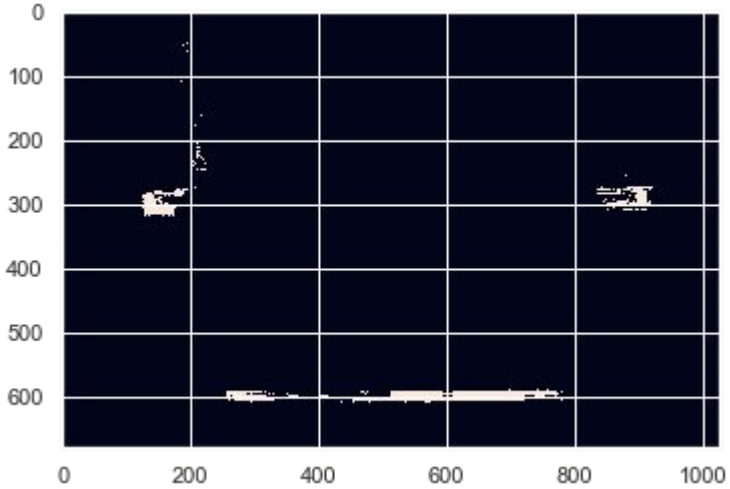
Prétraitement des images

- Egalisation :



Prétraitement des images

- Sans le background :

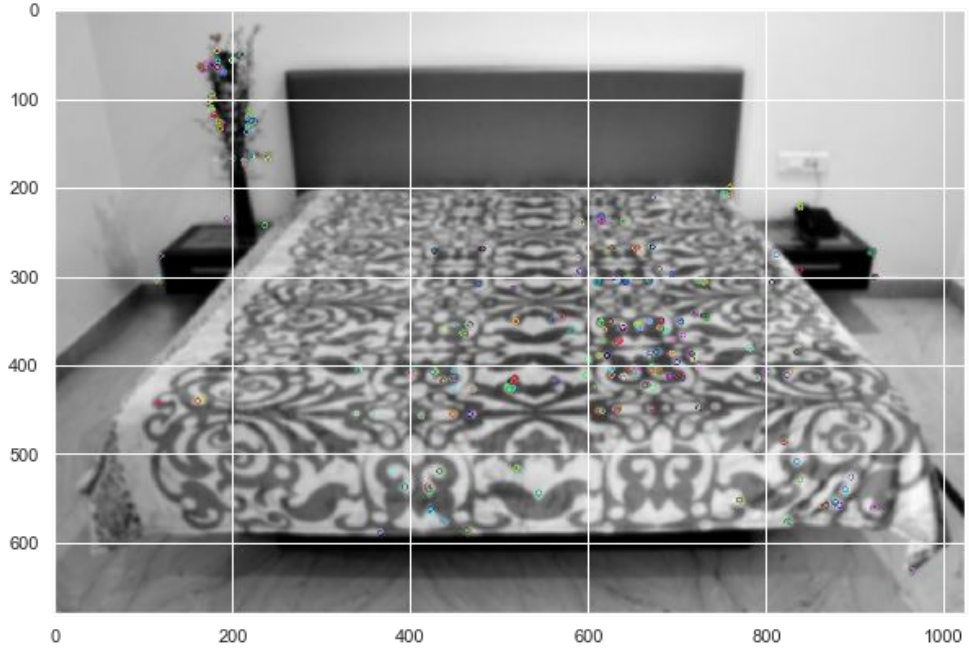


Etude des données visuelles

Classification des images

Classification des images

- On va utiliser ORB:



Classification des images

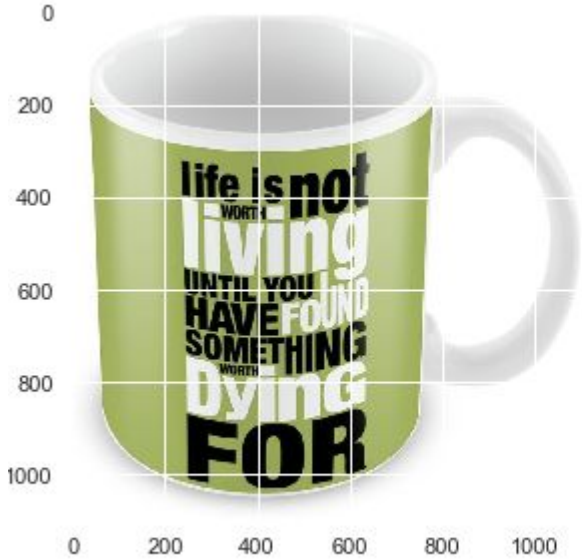
- Utilisation du MiniBatchKMeans pour générer l’histogramme (70 clusters)
- Etude sur les images processées vs non processées
- Utilisation d’un Random Forest pour la classification
  - Images non processées : training score = 0.38 vs testing 0.32
  - Images processées : training score = 0.38 vs testing 0.35

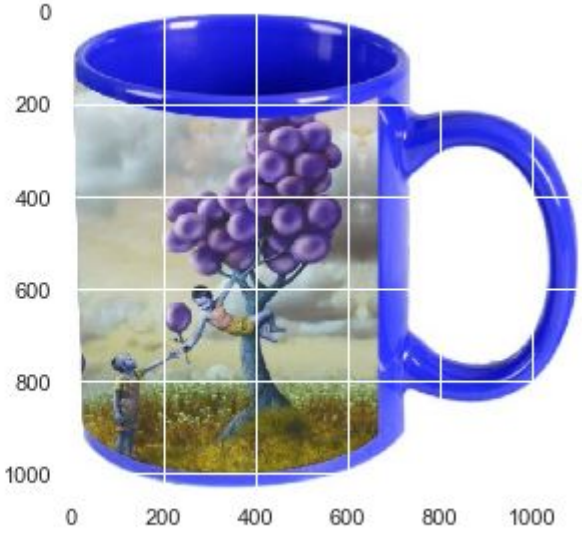
Classification des images

- Avec le Nearest Neighbors:

Classification des images

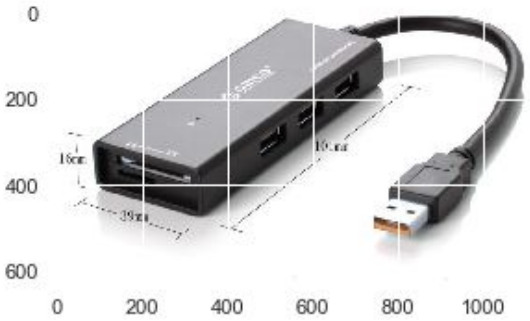
- Quelques images correspondent





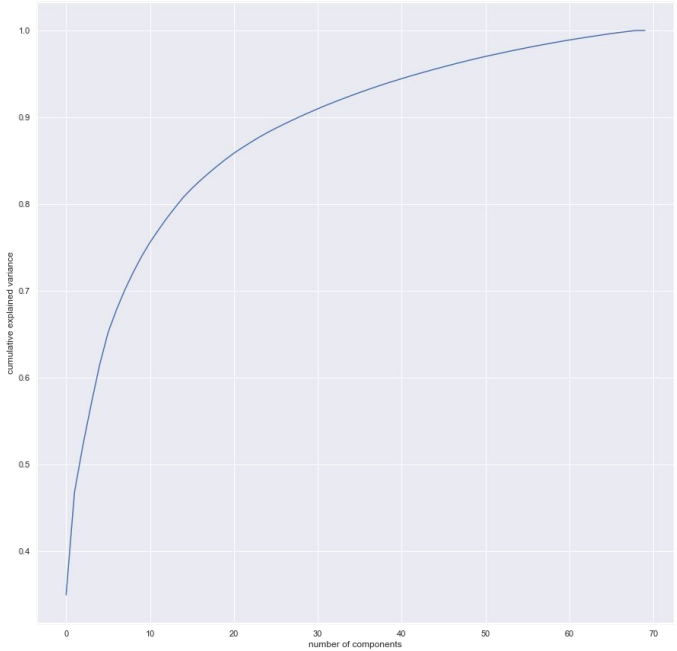
Classification des images

- D’autres pas du tout



Classification des images

- Réduction de dimension avec l’ACP



Classification des images

- Visualisation avec UMAP

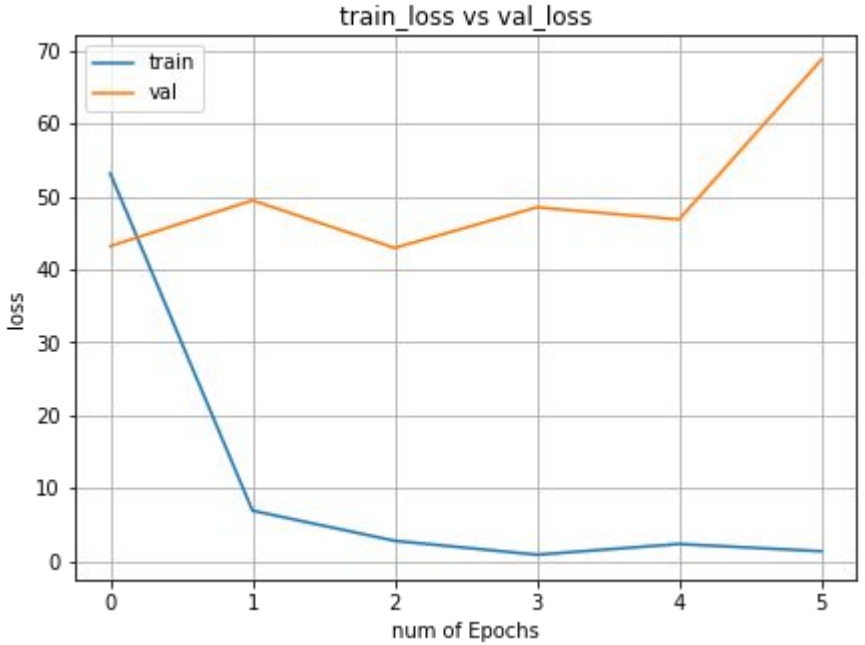
Etude des données visuelles

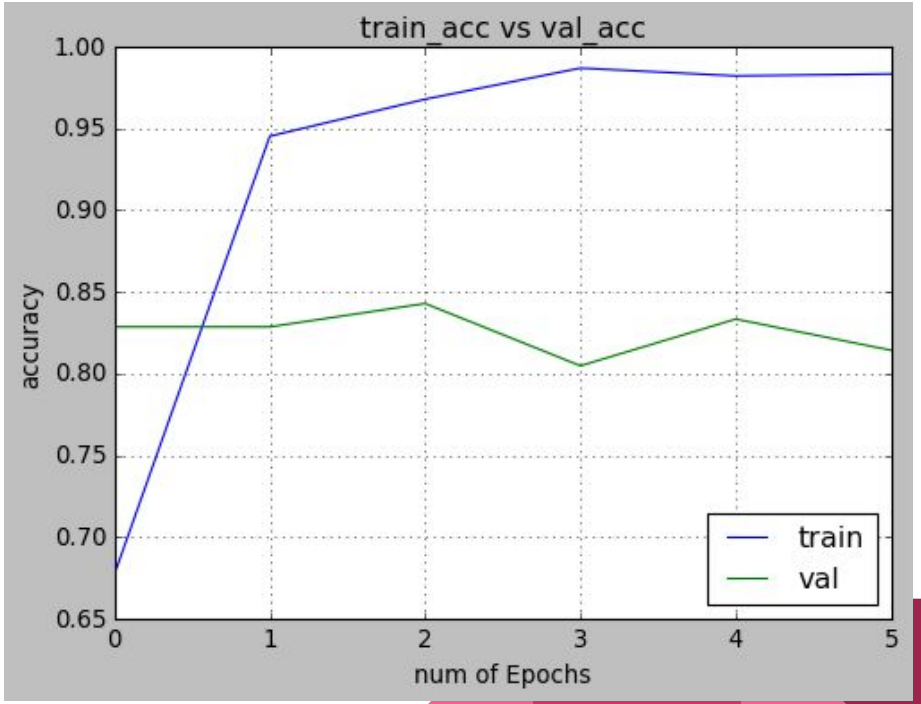
Utilisation d’un réseau de neurones

Utilisation d’un réseau de neurones

- Utilisation de ResNet50 :
  - On rajoute une couche Flatten avec une couche fully connected
    - activation_49 (Activation) (None, 7, 7, 2048) 0 add_16[0][0]
    - flatten (Flatten) (None, 100352) 0 activation_49[0][0]
    - dense_1 (Dense) (None, 7) 702471 flatten[0][0]
    - Total params: 24,290,183
    - Trainable params: 24,237,063
    - Non-trainable params: 53,120

Utilisation d’un réseau de neurones

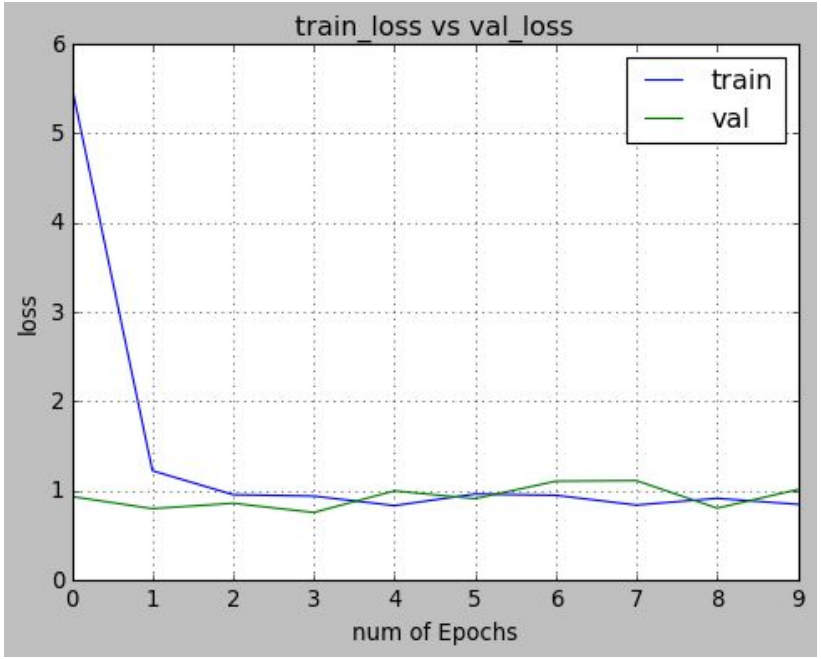


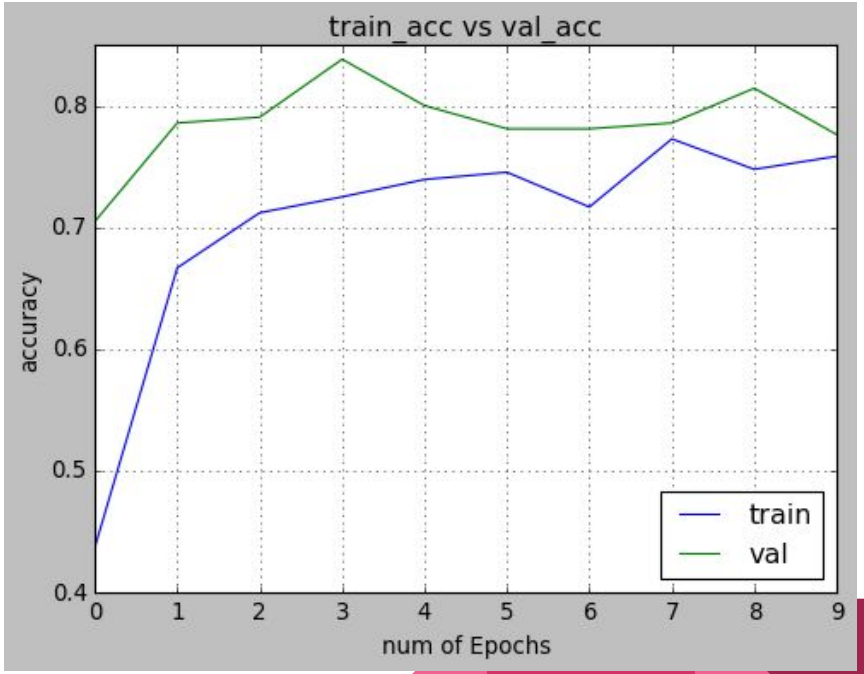


Utilisation d’un réseau de neurones

- Réduction de l’overfitting :
  - activation_147 (Activation) (None, None, None, 2 0 add_48[0][0]
  - global_average_pooling2d_2 (Glo (None, 2048) 0 activation_147[0][0]
  - fc-1 (Dense) (None, 512) 1049088 global_average_pooling2d_2[0][0]
  - dropout_1 (Dropout) (None, 512) 0 fc-1[0][0]
  - fc-2 (Dense) (None, 256) 131328 dropout_1[0][0]
  - dropout_2 (Dropout) (None, 256) 0 fc-2[0][0]
  - output_layer (Dense) (None, 7) 1799 dropout_2[0][0]
  - Total params: 24,769,927
  - Trainable params: 24,716,807
  - Non-trainable params: 53,120

Utilisation d’un réseau de neurones





Conclusion

Conclusion

- La classification textuelle pourrait s’auto-suffire
- La classification visuelle devrait passer par un CNN qu’on pourrait combiner avec l’analyse textuelle
- Pour une classifications moins générale, on peut utiliser les APIs existantes pour récupérer les données manquantes