Classifier automatiquement des biens de consommation

6

Place de marché

Parcours Data Scientist.

Présentation : Dai TENSAOUT

Plan

Contexte

- Place de marché : lancement d'une marketplace e-commerce.

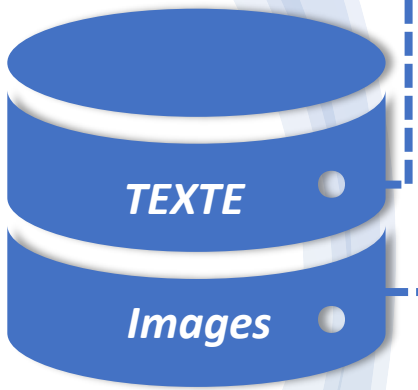
- Répondre à l'accroissement futur du nombre de catégorie de produits.
- Facilité la mise en ligne des nouveaux articles et fluidifier la recherche de produits.

- Faisabilité d'un moteur de classification automatique.
- Avec une précision suffisante.

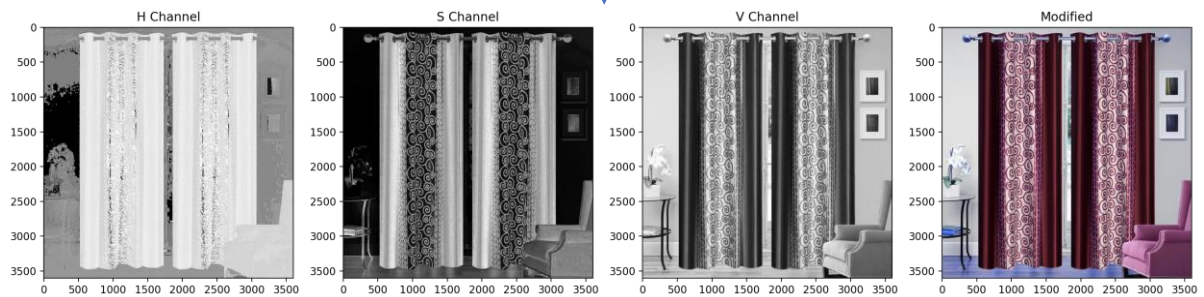
Données Clients

Une base de données composée de **1050** produits.

- **7** Catégories de produits.
- **150** Produits par catégorie.
- Un échantillon d'images : **PNG.**



****
Les 3 premiers éléments de la colonne : description
Key Features of Elegance Polyester Multicolor Abstract Eyelet Door Curtain Floral Curtain,Elegance Polyester Multicolor Abstrac t Eyelet Door Curtain (213 cm in Height, Pack of 2) Price: Rs. 899 This curtain enhances the look of the interiors.This curtain is made from 100% high quality polyester fabric.It features an eyelet style stitch with Metal Ring.It makes the room environmen t romantic and loving.This curtain is ant- wrinkle and anti shrinkage and have elegant apparance.Give your home a bright and mo dernistic appeal with these designs. The surreal attention is sure to steal hearts. These contemporary eyelet and valance curta ins slide smoothly so when you draw them apart first thing in the morning to welcome the bright sun rays you want to wish good morning to the whole world and when you draw them close in the evening, you create the most special moments of joyous beauty gi ven by the soothing prints. Bring home the elegant curtain that softly filters light in your room so that you get the right amo unt of sunlight.,Specifications of Elegance Polyester Multicolor Abstract Eyelet Door Curtain (213 cm in Height, Pack of 2) Gen eral Brand Elegance Designed For Door Type Eyelet Model Name Abstract Polyester Door Curtain Set Of 2 Model ID Duster25 Color M ulticolor Dimensions Length 213 cm In the Box Number of Contents in Sales Package Pack of 2 Sales Package 2 Curtains Body & Des ign Material Polyester



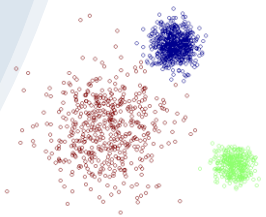
Méthodologie

Prétraitements

Extraction Features

Visualisation

Clustering



Méthodologie

Prétraitements

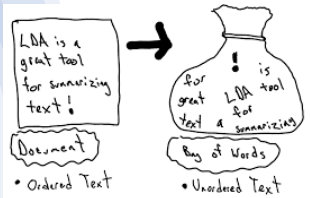
NLTK

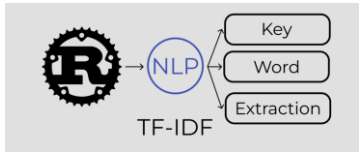
+

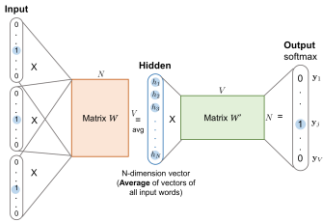
Tokenisation
StopWords
Ponctuations

Alpha numeric
Lemmatisation

Extraction Features



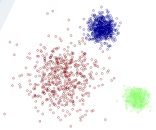




Visualisation

ACP

T-SNE



Clustering : KMeans

Méthodologie

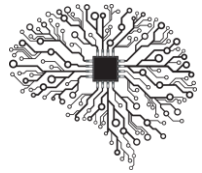
Prétraitements

OpenCV +

Resizing
Reshape
Preprocess_input

Extraction Features

Bag of Images
SIFT

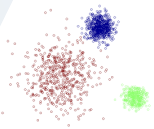


VGG16

Visualisation

ACP

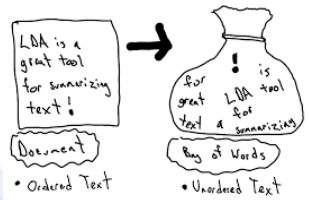
T-SNE

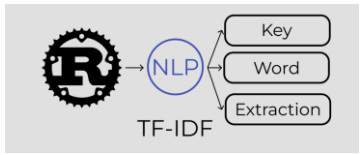


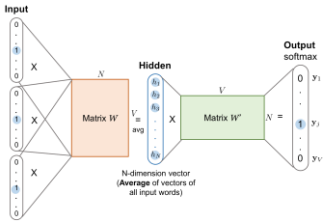
Classification supervisée

Résultats

Extraction Features







- ❑ **CountVectorizer** : (1050 , 4250)
- ❑ **TfidfVectorizer** : (1050 , 4250)

- ❑ **Word2Vec** : (1050 , 300)
- ❑ **BERT** : (1050 , 768)

- ❑ **USE** : (1050 , 512)

1. Fine Tuning T-sNE
2. Calcul ARI et T-sNE

Visualisation

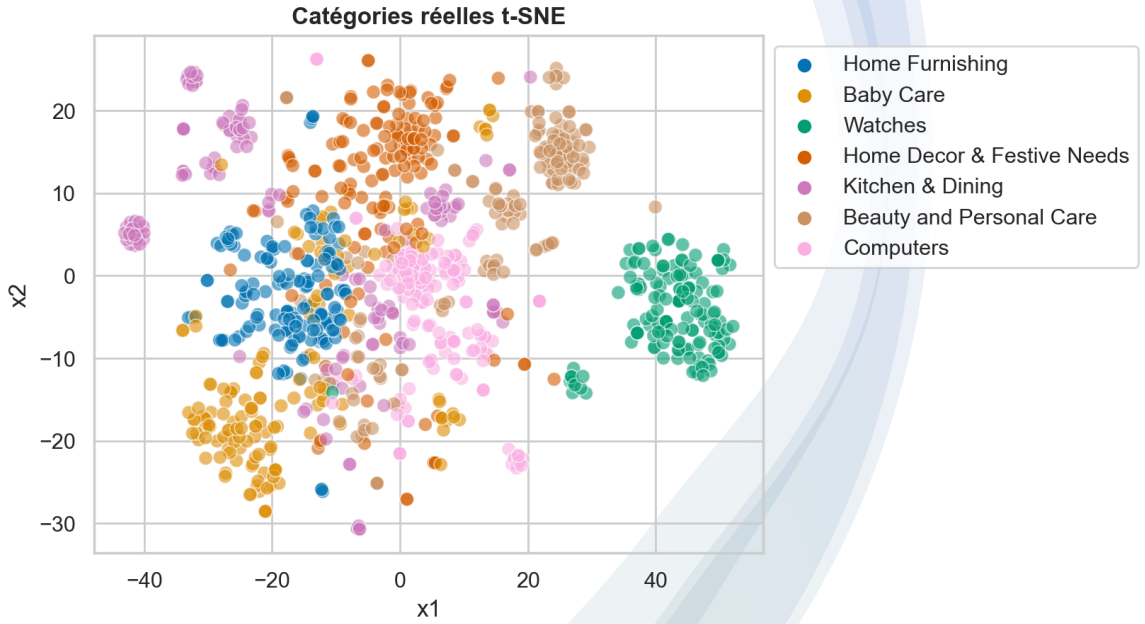
Résultats

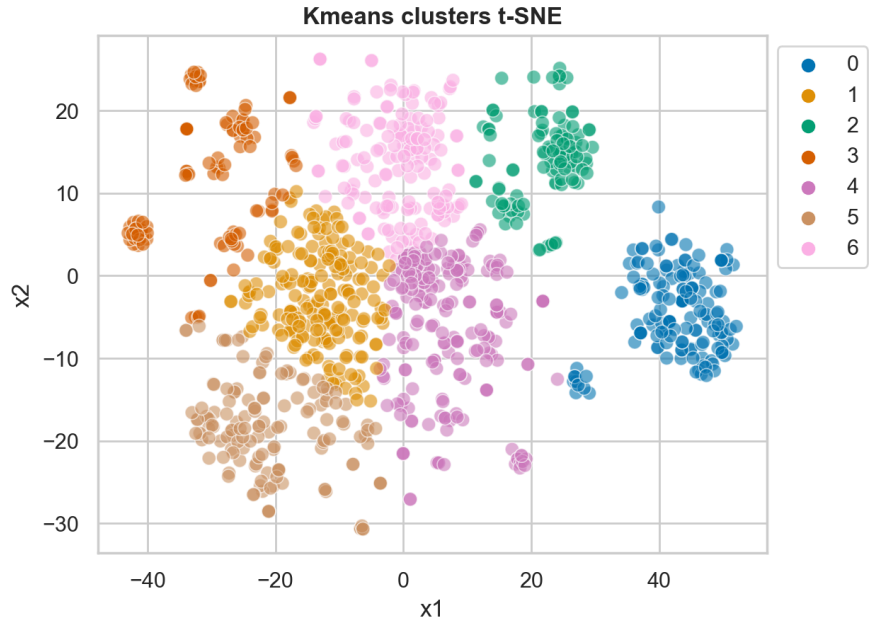
Visualisation

KMeans

T-SNE

❑ **CountVectorizer** : ARI = 51%





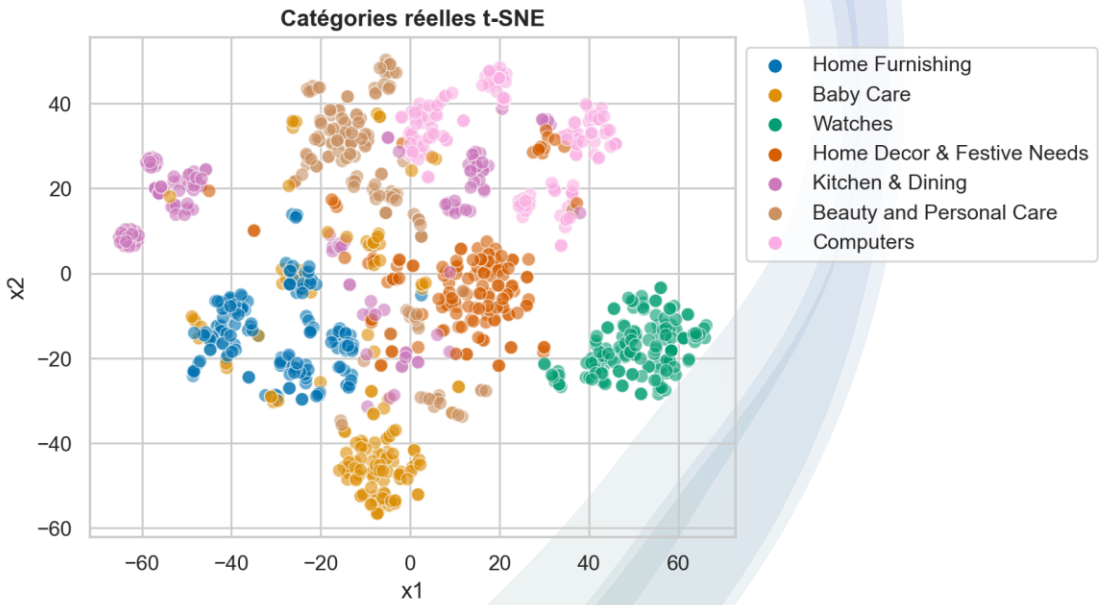
Résultats

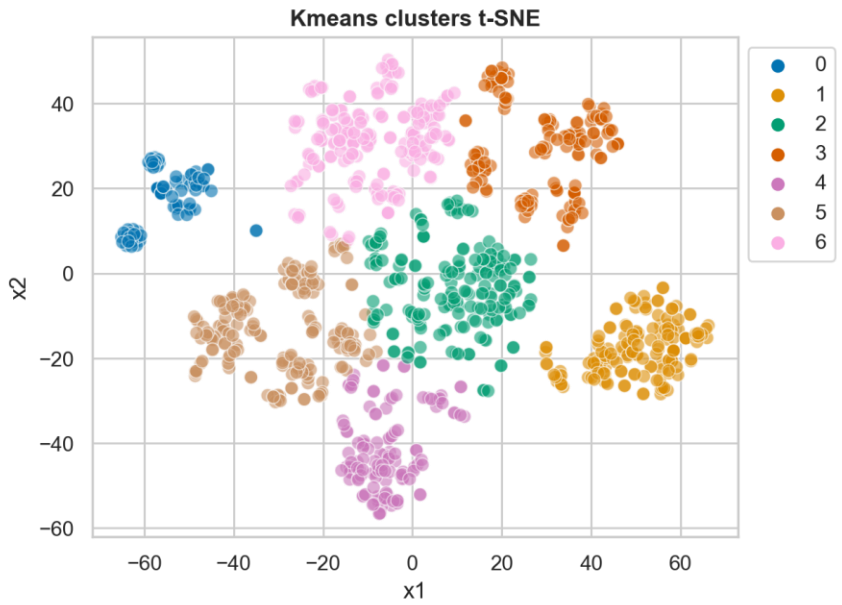
Visualisation

KMeans

T-SNE

❑ **TfidfVectorizer** : ARI = 55%





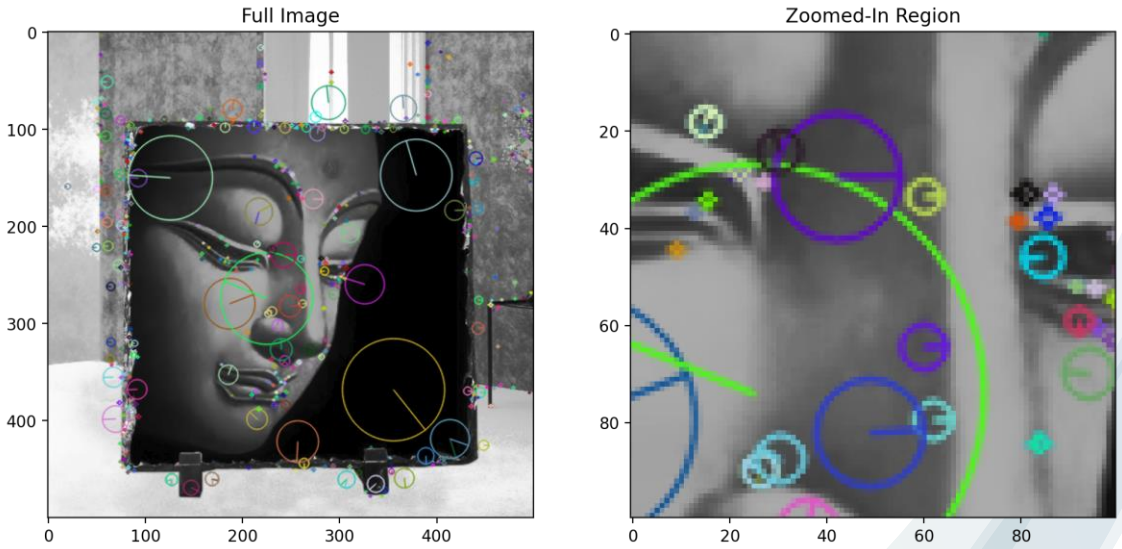
Résultats

Extraction Features

Bag of Images
SIFT

VGG16

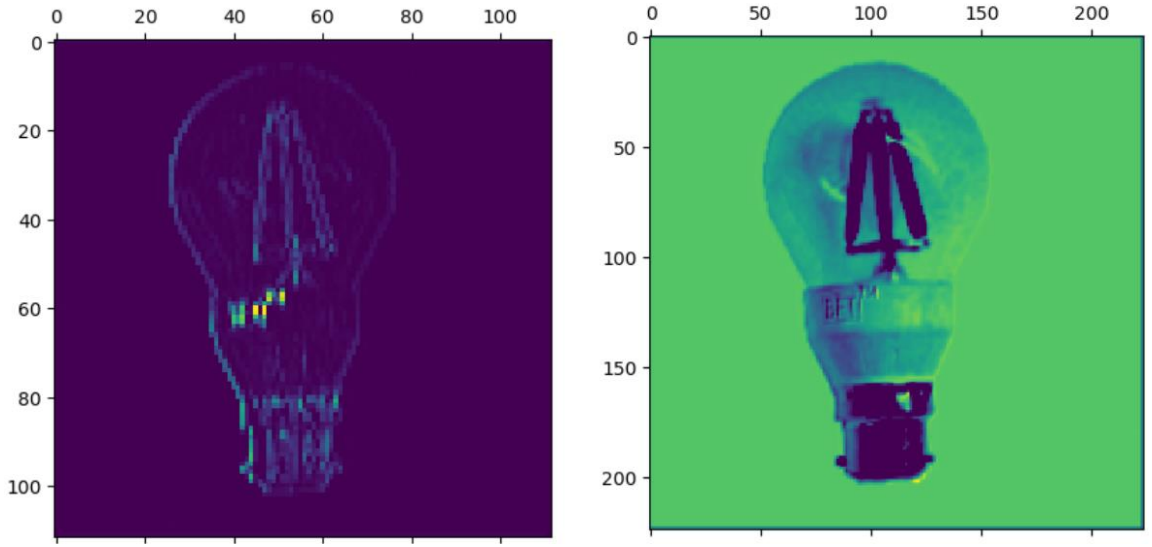
SIFT



Nombre de descripteurs : (810592, 128)

(1050, 900)

VGG16



(1050, 25088)

Résultats

Visualisation

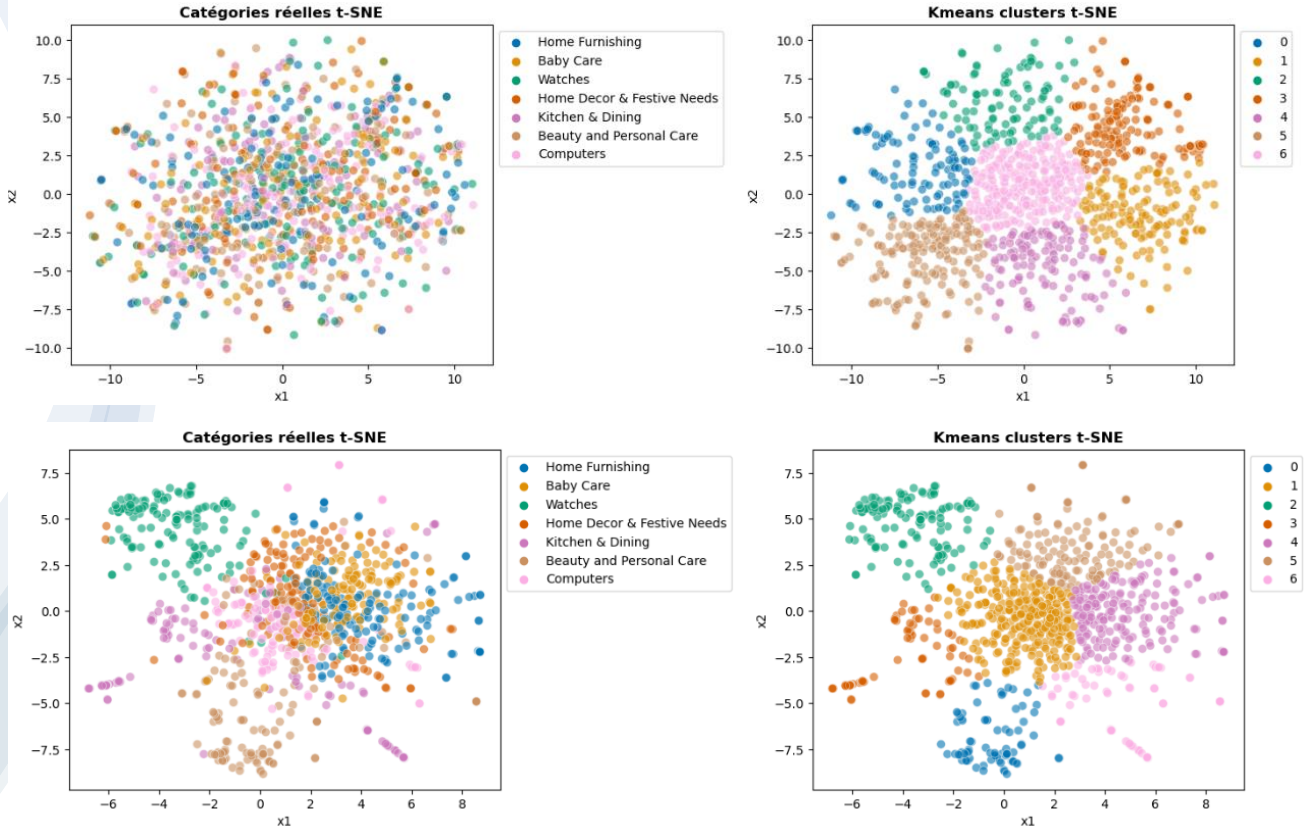
ACP

T-SNE

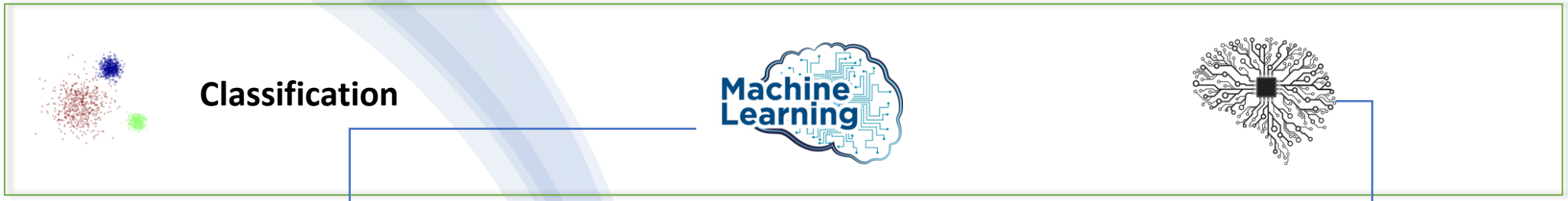
KMeans

❑ **SIFT** : ARI = -0,001 %

❑ **vgg16** : ARI = 32 %



Classification Supervisée : Résultats



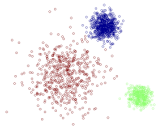

```
X_train :  (787, 940)
y_train :  (787,)

X_test :  (263, 940)
y_test :  (263,)
```

```
X_train :  (787, 224, 224, 3)
y_train :  (787, 7)
```

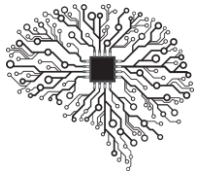
Model	F1_Score	Accuracy_Training	Accuracy_Testing	Time Taken
DecisionTreeClassifier	0.649468	1.000000	0.650190	0.495034
AdaBoostClassifier	0.592949	0.663278	0.623574	2.891153
LogisticRegression	0.493679	1.000000	0.490494	0.244920
KNeighborsClassifier	0.034702	0.143583	0.140684	0.081549

- **Approche :** préparation initiale des images
- **Approche :** par dataset Tensorflow

Classification Supervisée : Résultats



Classification



- **Approche :** préparation initiale des images

Preprocessing des images

```
path = "data/source/P6_images_data_set/"
img_name_col = data['image']
images = image_preprocessing_for_cnn(path, img_name_col)
```

train_test_split 25% -75%

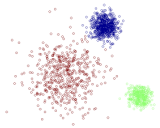
```
X_train :  (787, 224, 224, 3)
y_train :  (787, 7)
```

Entrainement d'un modèle vgg16 et évaluation

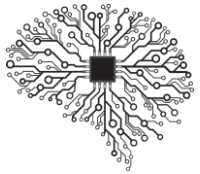
```
# Entraîner sur les données d'entraînement (X_train, y_train)
with tf.device('/gpu:0'):
    history1 = model1.fit(X_train, y_train, epochs=250, batch_size=64,
                          callbacks=callbacks_list, validation_data=(X_val, y_val), verbose=1)
```

```
25/25 [==============================] - 10s 150ms/step - loss: 0.0074 - accuracy: 0.9987
Training Accuracy: 0.9987293481826782

9/9 [==============================] - 1s 140ms/step - loss: 1.0385 - accuracy: 0.8099
Validation Accuracy:  0.8098859190940857
```

Classification Supervisée : Résultats



Classification



❑ **Approche :** par dataset Tensorflow

Préparation des dataset

Split des Image en Train Set et Test Set

Image_dataset_from_directory

```
Found 735 files belonging to 7 classes.
Using 515 files for training.

Found 735 files belonging to 7 classes.
Using 220 files for validation.

Found 315 files belonging to 7 classes.
```

Entrainement d'un modèle vgg16 et évaluation

```
batch_size = 32
with tf.device('/gpu:0'):
    history3 = model3.fit(dataset_train,
                          validation_data=dataset_val,
                          batch_size=batch_size, epochs=50, callbacks=callbacks_list, verbose=1)
```

```
# Score du dernier epoch

loss, accuracy = model3.evaluate(dataset_test, verbose=True)
print("Training Accuracy   : {:.4f}".format(accuracy))
print()
```

```
10/10 [==============================] - 8s 719ms/step - loss: 1.4427 - accuracy: 0.7683
Training Accuracy   : 0.7683
```

Conclusions

- En dépit du manque de données, certaines techniques d'extraction de *features* permettent bien de détecter le contexte dans le cas du texte, et les différents descripteurs dans le cas des images.
- Entrainer les *classifiers* sur des données plus conséquentes, afin d'augmenter la précision de la classification.
- Nous confirmons bien la possibilité de construire un moteur de classification automatique de produits.

Classifier automatiquement
des biens de consommation

Parcours Data Scientist.

Merci de votre attention

Présenté par Mr Dai TENSAOUT