Fuzzy matching on big-data

An illustration with scanner data and crowd-sourced nutritional data

Lino Galiana (1) ; Milena Suarez-Castillo (2)
(1) D2E, (2) SSP-lab
31/03/2022

Problématique (1/2)

- Alimentation affecte variables de **santé publique** ayant un effet de premier ordre sur bien-être :
  - obésité
  - mortalité différentielle
- **Facteurs sociaux** et **territoriaux** interagissent
  - Interaction avec les inégalités socioéconomiques: Caillavet et al. (2020)
  - Malgré réglementations (ex: nutriscore), beaucoup d'asymétries d'information
- Alcott et al. (2019) proposent une décomposition entre effets:
  - d'offre (déserts alimentaires)
  - de demande (préférences pour la *junk food*)
- Commencer par documenter les inégalités de *"qualité"* de consommation
  - Avant d'essayer de distinguer effets offre et demande

Problématique (2/2)

- Au sein d'une classe de produit (ex: lasagnes), énormément d'**hétérogénéité** de *"qualité"*
  - Low-cost vs autres produits
  - Bio vs agriculture traditionnelle
- Besoin de **données très fines** pour distinguer les qualités nutritionnelles (Caillavet et al. 2020)
- D'où l'utilisation de **sources de collectes automatisées**:
  - Données d'enquête (notamment Budget des Familles) serviront à vérifier la cohérence voire caler
  - Autres sources administratives pour contrôler la qualité des données

Enjeux

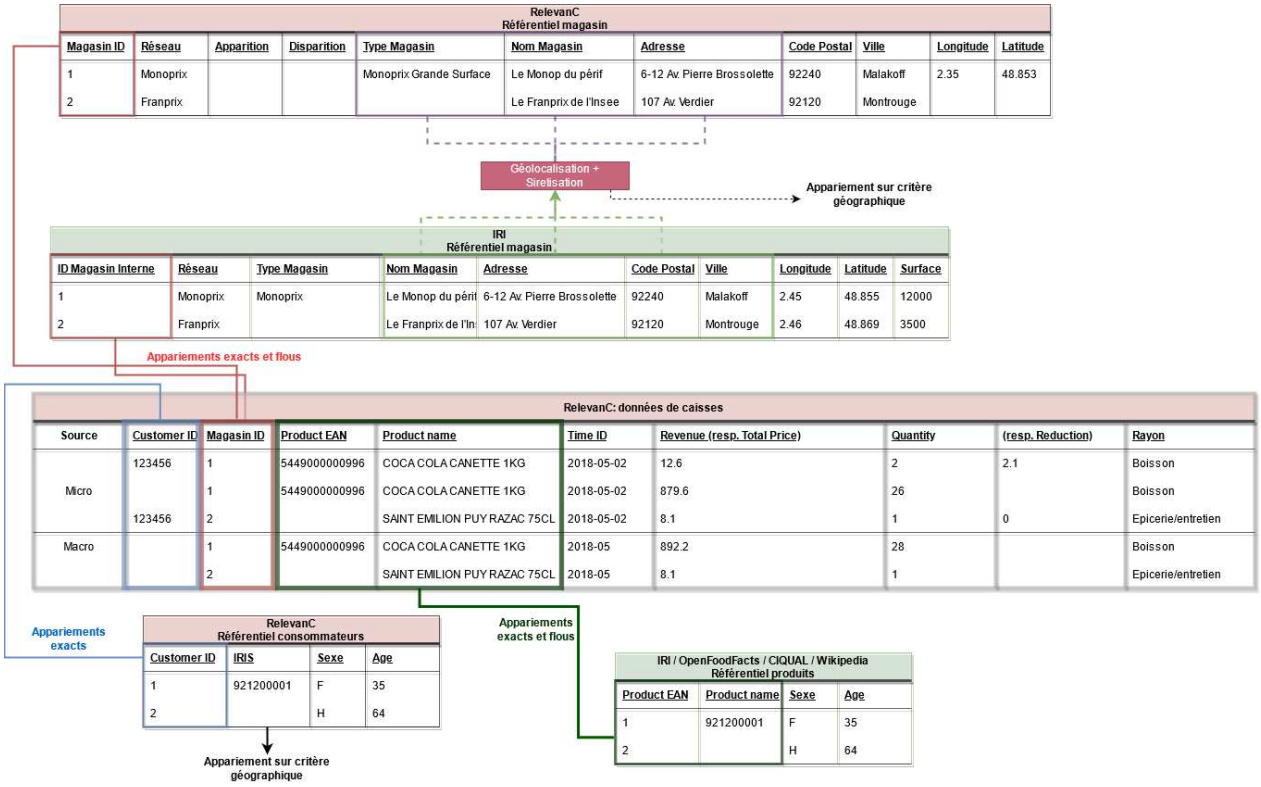
- Besoin d'**enrichissements multiples**:
  - Sur la dimension `géographique` ;
  - Sur la dimension `produit`
- Méthode des **appariements flous**:
  - Compenser absence d'identifiants pour appariements exacts
  - Noms produits
  - Adresse magasins
- **Catégorisation** pour réduire le nombre de paires
  - COICOP
  - Département

> ### Remarque
>
> Approche généralisable à de nombreux problèmes de la statistique publique

RelevanC

- Données de caisses du groupe Casino
  - Casino, Monoprix, Franprix au *niveau magasin* et *cartes fidélité*
  - Leader Price et des petites chaînes (stations services, Sherpa...) au *niveau magasin* exclusivement
- ≈ 11 000 magasins
- ≈ 250 000 produits

RelevanC


RelevanC
Référentiel magasin
Magasin ID | Réseau | Apparition | Disparition | Type Magasin | Nom Magasin | Adresse | Code Postal | Ville | Longitude | Latitude
1 | Monoprix | | | Monoprix Grande Surface | Le Monop du périf | 6-12 Av. Pierre Brossolette | 92240 | Malakoff | 2.35 | 48.853
2 | Franprix | | | | Le Franprix de l'Insee | 107 Av. Verdier | 92120 | Montrouge | | |
Géolocalisation + Sirétisation
Appariement sur critère géographique
IRI
Référentiel magasin
ID Magasin Interne | Réseau | Type Magasin | Nom Magasin | Adresse | Code Postal | Ville | Longitude | Latitude | Surface
1 | Monoprix | Monoprix | Le Monop du périt | 6-12 Av. Pierre Brossolette | 92240 | Malakoff | 2.45 | 48.855 | 12000
2 | Franprix | | Le Franprix de l'In | 107 Av. Verdier | 92120 | Montrouge | 2.46 | 48.869 | 3500
Appariements exacts et flous
RelevanC: données de caisses
Source | Customer ID | Magasin ID | Product EAN | Product name | Time ID | Revenue (resp. Total Price) | Quantity | (resp. Reduction) | Rayon
123456 | 1 | 5449000000996 | COCA COLA CANETTE 1KG | 2018-05-02 | 12.6 | 2 | 2.1 | Boisson
Micro | 1 | 5449000000996 | COCA COLA CANETTE 1KG | 2018-05-02 | 879.6 | 26 | | Boisson
123456 | 2 | | SAINT EMILION PUY RAZAC 75CL | 2018-05-02 | 8.1 | 1 | 0 | Epicerie/entretien
Macro | 1 | 5449000000996 | COCA COLA CANETTE 1KG | 2018-05 | 892.2 | 28 | | Boisson
2 | | SAINT EMILION PUY RAZAC 75CL | 2018-05 | 8.1 | 1 | | Epicerie/entretien
Appariements exacts
RelevanC
Référentiel consommateurs
Customer ID | IRIS | Sexe | Age
1 | 921200001 | F | 35
2 | | H | 64
Appariement sur critère géographique
Appariements exacts et flous
IRI / OpenFoodFacts / CIQUAL / Wikipedia
Référentiel produits
Product EAN | Product name | Sexe | Age
1 | 921200001 | F | 35
2 | | H | 64

`OpenFoodFacts` (openfoodfacts.org)

- Base de données contributive et open-source (alternative à `Yuka`)
  - Actualisée en continu ;
  - Mise à disposition de csv de manière quotidienne ;
  - Plusieurs API
- ≈ 2 millions de produits
- Beaucoup d'infos disponibles
  - *Scores de qualité agrégés* : nutriscore, score NOVA, écoscore
  - *Infos nutritionnelles*: calories, glucides, lipides, etc.
  - *Infos sur produit*: packaging, etc.
- *Challenge*: qualité et taux de complétude variable
  - Valeurs manquantes
  - Erreurs de remplissage et fautes d'ortographe

CIQUAL (ciqual.anses.fr)

- Base de données élaborée par l'Anses
  - **Produits standardisées**
  - Beaucoup d'infos disponibles
  - *Challenge*: meilleure qualité mais produits standardisés
- Globalement même information qu'OpenFoodFacts :
  - Mais, perte d'une dimension dans l'hétérogénéité produit
- Besoin d'enrichir certaines classes de produits:
  - Faible variabilité nutritionnelle
  - Forte hétérogénéité des labels des sous-classes: marques, domaines, goûts...

Dictionnaires de produits via Wikipédia

```
'VINS':  {
    'ciqual' : 'Vin rouge',
    'wiki':  ['Vin AOC en France par région','Vin_français']
},
'FROMAGES': {
    'ciqual': 'Fromage (aliment moyen)',
    'wiki': ['Marque_de_fromage_en_France','Fromage_français']
},
'CHIPS': {
    'ciqual' : 'Chips de pommes de terre nature ou aromatisées, standard',
    'wiki': ['Marque_de_chips']
},
```

Résultat de l'API de MediaWiki


```
{
    "query": {
        "categorymembers": [
            {
                "pageid": 14025683,
                "ns": 0,
                "title": "Cheetos"
            },
            {
                "pageid": 4092151,
                "ns": 0,
                "title": "Doritos"
            },
```


Indexation de produits composites

Produits "Cheetos" (Désignation Wikipédia)
Caractéristiques Nutrionnelles: Celles de "Chips de pommes de terre nature ou aromatisées, standard" issue de Ciqual

Particulièrement utiles pour les boissons alcoolisées

- Réutilisation d'un output d'un projet du SSP-Lab (`PAul-EAN`):
  - modèle de prédiction de la COICOP
  - entrainé sur *l'ensemble* des données de caisse
- Pour deux usages :
  - COICOP prédite servira de variable de blocage
  - Création d'une *distance* ad-hoc entre libellés:
- Via un plongement de mots (`word embedding`)
  - initialisé sur le plongement de mots issu de ce modèle

Schéma macro du pipeline

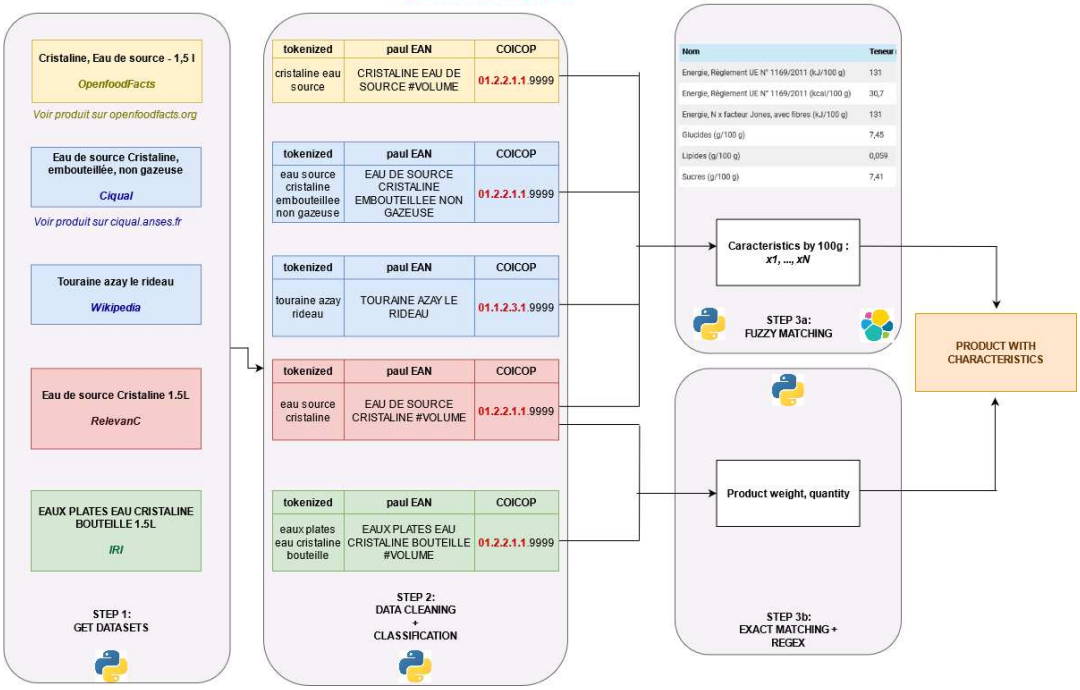
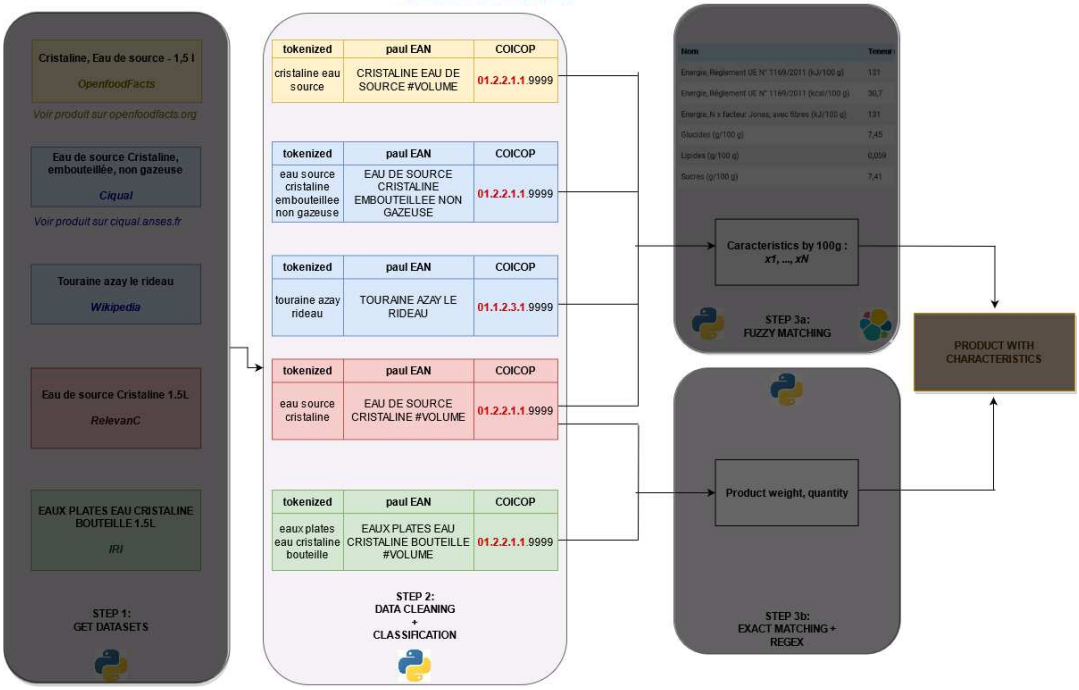


Schéma macro du pipeline



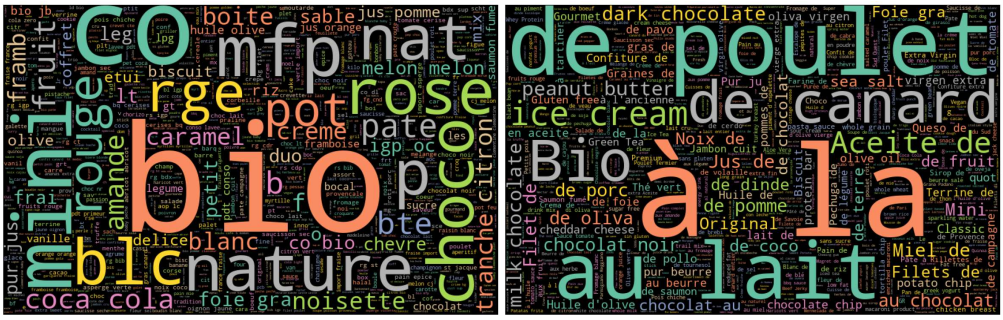
- Réduire le bruit dans les données ;
- Harmoniser les jeux de données ;
- Détecter des produits non alimentaires malgré filtre rayons.

Détails

Word clouds RelevanC (left) and Open Food (right)

Deuxième étape: normalisation des noms de produits

☑ Réduire le bruit dans les données ; ☑ Harmoniser les jeux de données ; ☑ Détecter des produits non alimentaires malgré filtre rayons.



Word clouds RelevanC (left) and Open Food Facts (right)

Deuxième étape: normalisation des noms de produits

Examples of preprocessing output from our pipeline

Libellé d'origine	Libellé tokenisé	Libellé classification
Cristaline, Eau de source - 1,5 l	cristaline eau source	CRISTALINE EAU DE SOURCE #VOLUME
Eau de source Cristaline, embouteillée, non gazeuse	eau source cristaline embouteillee non gazeuse	EAU DE SOURCE CRISTALINE EMBOUTEILLEE NON GAZEUSE
Touraine azay le rideau	touraine azay rideau	TOURAINE AZAY LE RIDEAU
Eau de source Cristaline 1.5L	eau source cristaline	EAU DE SOURCE CRISTALINE #VOLUME
EAUX PLATES EAU CRISTALINE BOUTEILLE 1.5L	eaux plates eau cristaline bouteille	EAUX PLATES EAU CRISTALINE BOUTEILLE #VOLUME
Eau de source Cristaline 6X1.5L	eau source cristaline	EAU DE SOURCE CRISTALINE #LOT #VOLUME

- Chercher un nom de produit proche dans $\approx$ 2 millions est une tâche excessivement complexe...
    - ... surtout quand on doit le faire 250 000 fois
    - Equivalent à un produit cartésien de 10^{13} opérations (nombre de cellules dans le corps humain)
- Trouver une **variable de blocage** pour n'associer qu'une partie des paires entre-elles:
    - Améliore la pertinence de la recherche
    - Réduit drastiquement la complexité du problème
- Pas de variable commune dans nos différentes sources:
    - Rayon: faible qualité, pas correspondance entre sources
- Utilisation `COICOP`

- Réseau neurone entraîné sur données de caisses
  - Jamais accès aux données d'entraînement !
  - Utilisation des poids estimés grâce au binaire `fasttext`

Libellé d'origine	Libellé tokenisé	COICOP	Label
LE PANIER FAISSELLE BIO 4X100G	panier faisselle bio	01.1.1.5.1.9999	Bread and cereals
NAVARIN AGNEAU 1,2KE	navarin agneau	01.1.2.8.3.0010	Meat
SAUMON SAUVAGE PROV.MSC 330G BQ	saumon sauvage prov msc	01.1.3.6.1.9999	Fish and shellfish
ABRICOT 35/45 BQ 1KG	abricot	01.1.6.3.1.0005	Fruits
POTE AUVERGNATE 400G	pote auvergnate	01.1.7.6.1.9999	Vegetables
MIEL ROMARIN HAUTE VALLES 480G	miel romarin haute valles	01.1.8.4.1.0016	Confectionery and frozen products
ENTREMETS CITRON MERINGUE 6P 500G	entremets citron meringue	01.1.8.5.1.9999	Confectionery and frozen products
HERBE MENTHE POT	herbe menthe pot	01.1.9.2.1.0017	Salt, spices and sauces
COCA-COLA ZERO PET 1.5LX6 CONT MAST	cocacola zero pet cont mast	01.2.2.2.1.0006	Other soft drinks
BISTROT DE FRANCE RS BIB 5L	bistrot france rs bib	02.1.2.1.1.0004	Wines, ciders and champagne

Pipeline général

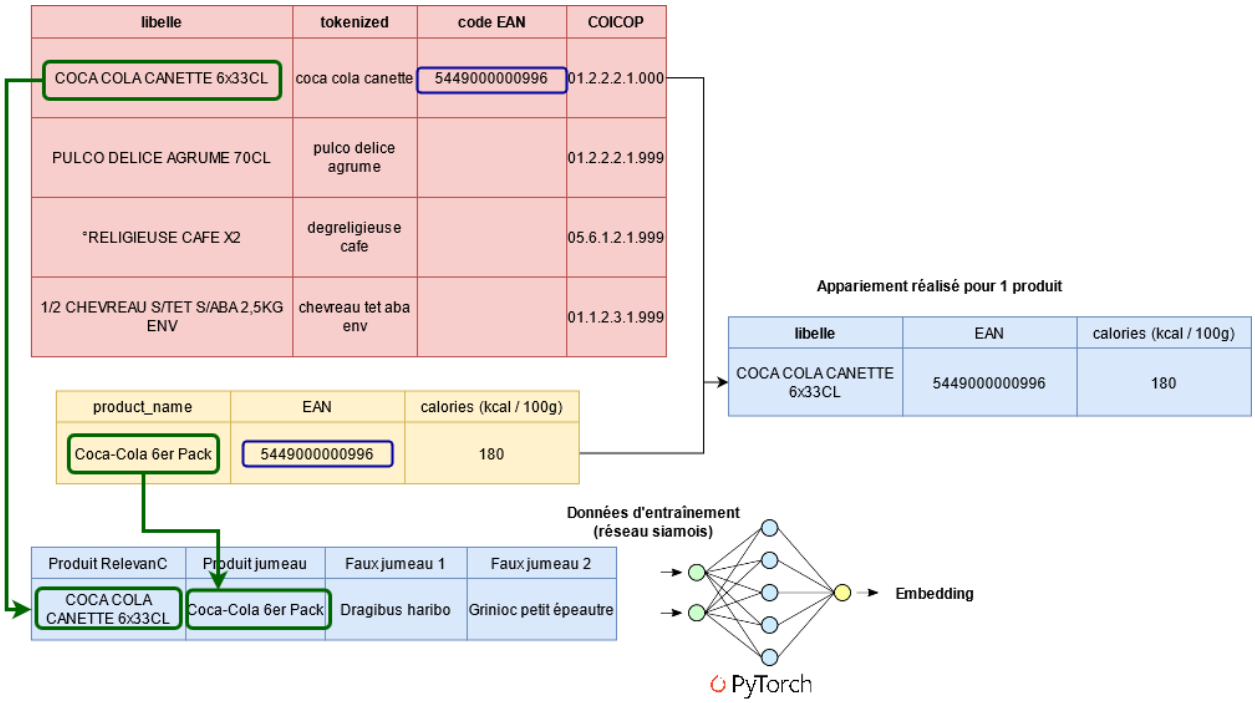
- On recherche dans `OpenFood` / `CIQUAL` un produit des données de caisses
- Une procédure pour aller de l'appariement le plus certain au plus incertain:
  1. **Appariement EAN** si disponible dans `OpenFood`
  2. **Appariement flou** dans les produits `OpenFood` ayant le même code `COICOP`
  3. **Appariement flou** dans tous les produits `OpenFood`
  4. **Appariement flou** dans tous les produits `CIQUAL` (enrichie avec le dico `Wikipedia`)
- **Critères de validation** conservateurs pour exclure faux positifs

Pipeline général

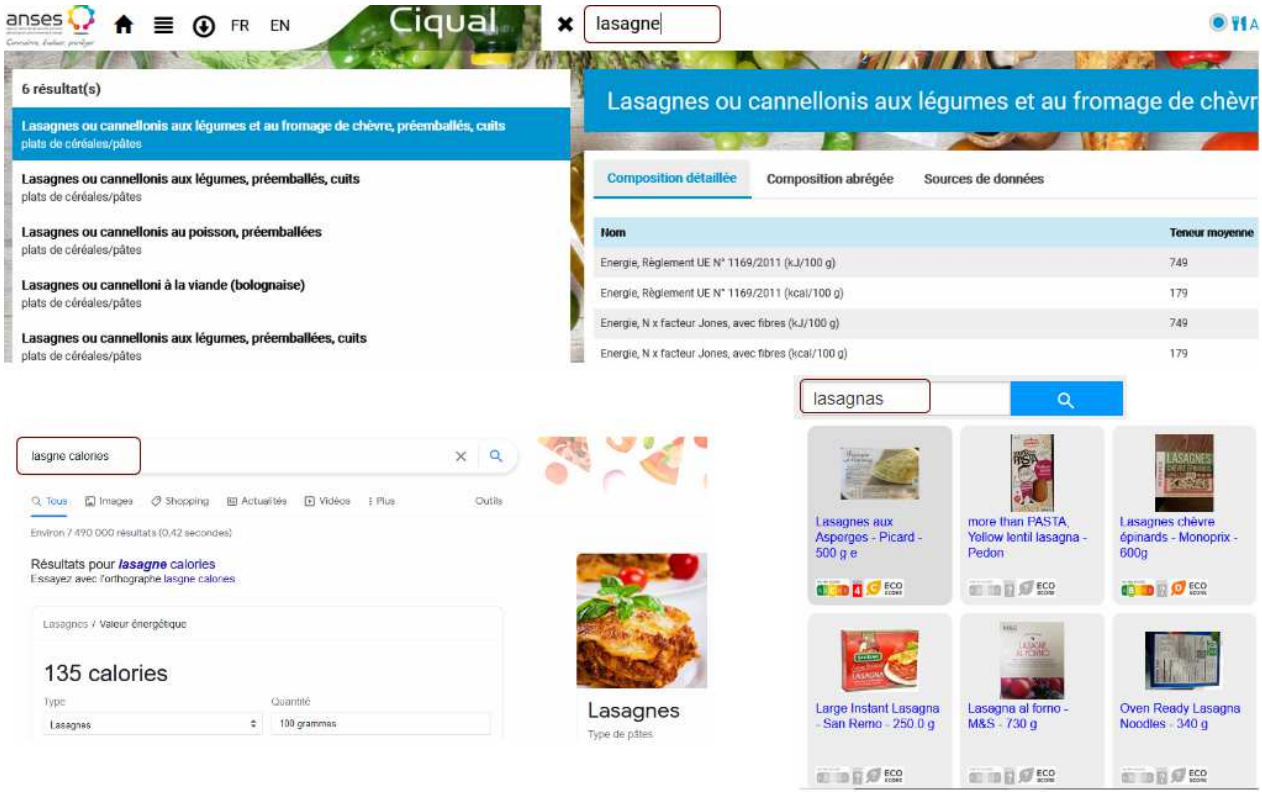
- Code `Python` pour gérer ce *pipeline*

- Package `foodbowl`
  - Gère la connexion à `S3` et `Elastic`
  - Mise en forme des requêtes, récupération des échos...
  - Validation des *outputs* (distance textuelle et potentiellement réseau de neurone `PyTorch`)
  - Déploiement sous forme d'API grâce à `fastAPI` (à venir)

- EAN est un identifiant produit unique: appariement exact
- Pour la calorie: permet d'enrichir 46% des données `RelevanC`
- Appariement servira de base d'entraînement pour le réseau siamois



Objectif du fuzzy matching



Pourquoi `ElasticSearch` ?

- Liberté dans la recherche:
  - Mélange critères distances textuelle pure (type Levensthein)...
  - ... avec d'autres critères (ngrams...)
- Extrêmement performant:
  - Parallélisation des recherches, recherches multiples, asynchrones, etc.
  - Recherches optimisées

Implementation	Speed (sec.)
Mixed implementation	
`ElasticSearch` [a]	0.92
Python	
`Rapidfuzz` [b]	97.20
`Fuzzywuzzy`	99.90

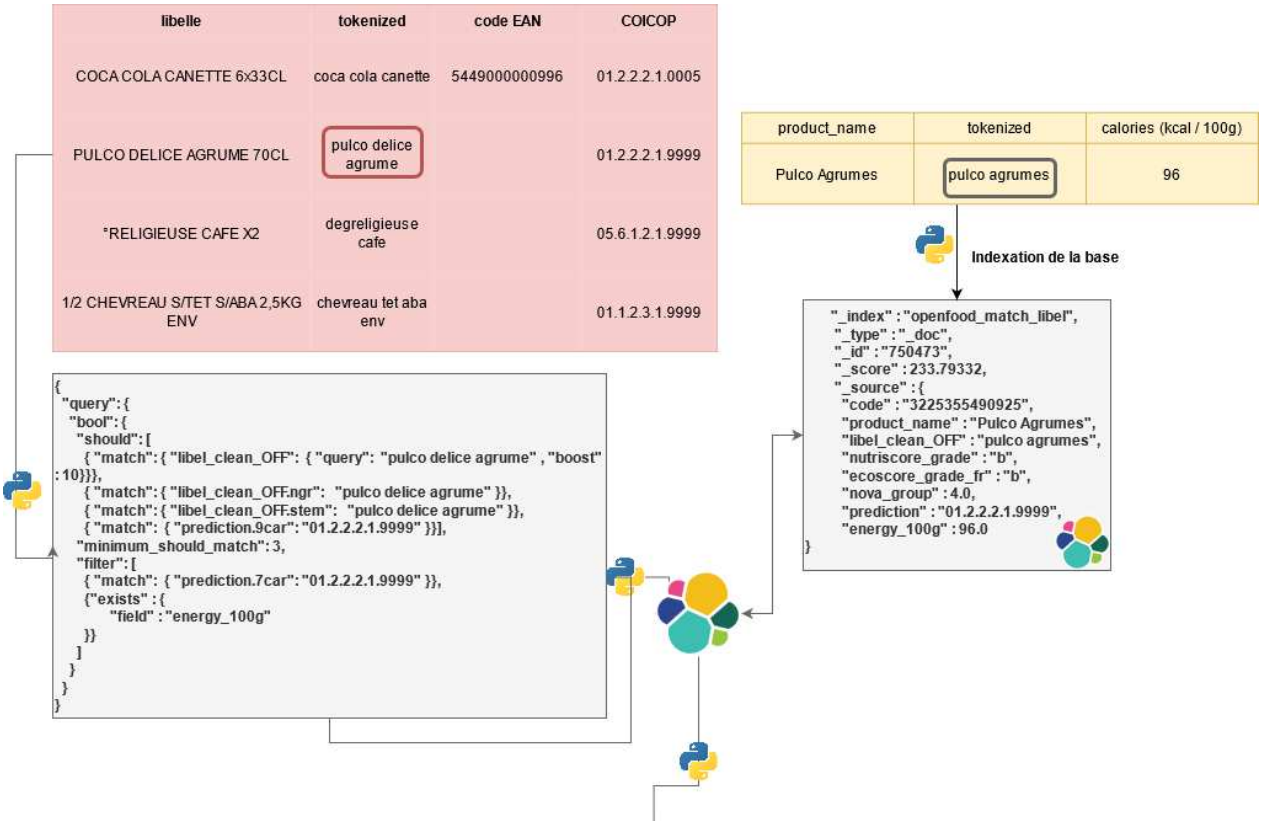
[a] `ElasticSearch` time includes the loss of time of sending data from Python to ElasticSearch server and bringing back data from the best match

[b] `Rapidfuzz` uses, behind the stage, `C` rather than relying on a slow pure `Python` implementation

`ElasticSearch` comment ?

- *Open-source*, possibilité d'avoir des instances internes
  - SSP-Cloud
  - Instance Kube interne
- Livré avec interface graphique `Kibana` :
  - Pratique pour *monitoring*
  - Mieux vaut passer par `Python` pour **indexation** & **requêtage**
- Package officiel `elasticsearch` pour intégration à `Python`
  - Repose sur le format JSON (pratique en `Python` !)

ElasticSearch dans notre *pipeline*



ElasticSearch avec

- Connexion simple:

```
from elasticsearch import Elasticsearch
es = Elasticsearch([{'host': HOST, 'port': 9200}], http_compress=True, t
```

- Création d'un *index* à partir d'un `DataFrame` :

```
from elasticsearch.helpers import parallel_bulk
es.indices.create(index=index_name, body=mapping)
deque(parallel_bulk(client=es, actions=gen_dict_from_pandas(index_name,
```

- Envoi et récupération de plusieurs requêtes :

```
es.msearch(body=req, max_concurrent_searches=1000)
```

Exemple de requête

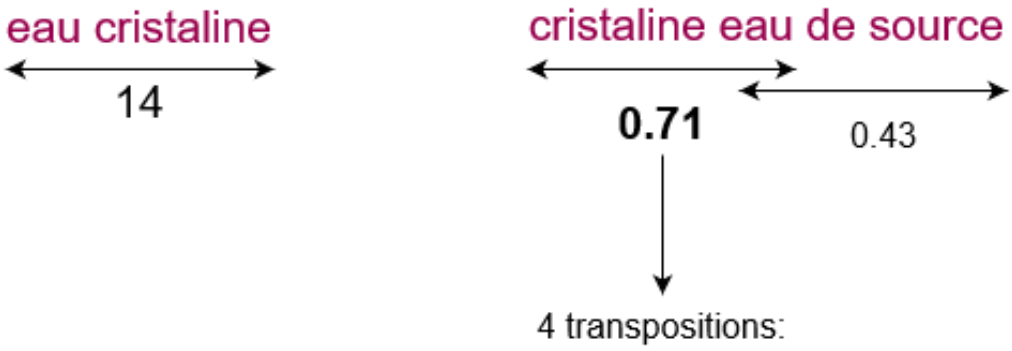
```
{
  "query": {
    "bool": {
      "should": [
        { "match": { "libel_clean_OpenFoodFact":  { "query":  "pulco del
        { "match": { "libel_clean_OpenFoodFact.ngr":   "pulco delice agr
        { "match": { "libel_clean_OpenFoodFact.stem":   "pulco delice ag
        { "match":  { "COICOP.6digits": "01.2.2.2.1.9999" }}],
      "minimum_should_match": 3,
      "filter": [
        { "match":  { "COICOP.5digits": "01.2.2.2.1.9999" }},
        {"exists" : {
               "field" : "energy_100g"
        }}
      ]
    }
  }
}
```

`ElasticSearch` avec 🐍: retour d'expérience

- Pas besoin d'avoir la certification *Elastic Engineer* (https://www.elastic.co/fr/training/elasticsearch-engineer) pour gagner à l'utiliser
- Gains de performance importants, même avec une requête sous-optimale
    - Requêtes asynchrones à creuser
- Gain en flexibilité à creuser
    - Exemple: étape de validation par distance Levensthein avant le renvoi d'écho serait plus performant

> **Remarque**
>
> Objectif: supprimer les faux positifs

- `rapidfuzz.fuzz.partial_ratio(s1, s2)` : meilleure similarité entre
  - la chaîne de caractère la plus courte `s1`, de longueur `n`
  - les `n`-grammes de `s2` 😱
- Similarité : distance de Levenshtein (y compris transposition) normalisée par la longueur `n`
- **Critère conservateur:** validation paire si `partial_ratio(s1, s2) > 0.65`



problème?

La proximité de caractère (**proximité syntaxique**) n'implique pas nécessairement une **proximité sémantique**.

Examples of false positives under the Levenstein distance

RelevanC	Open Food Facts
oe poulet blanc fermier	poulet blanc fermier
tarte chevre epinar co	epinar
lapin or sous alu	oeufs sous alu

- Test d'un plongement de mots (`word embedding`):
  - associer à un libellé un vecteur dans $\mathbb{R}^N$
  - dont la position dans cette espace reflèterait un concept de produit
- En déduire une distance non exclusivement basée sur la chaîne de charactère

Principe

Exemples

Terme	Cinq libellés les proches voisins[a]
NUTELLA	NUTELLA, NUTELLA, CANISTRELLI, PATE A TARTINER CHOCOLAT NOISETTES
MIKADO	GLICO MIKADO, MIKADO KING MIKADO CHOCOLAT NOIR, CIGARETTES RUSSES, MIKADO 3X CH LAIT LOTX EA, OVOMALTINE
CROCODILES HARIBO	HARIBO TIRLIBIBI, BONBONS GELIFIES CROCODILES HARI HARIBO, BONBON DRAGIBUS, BONBONS LUTTI MINI, CHUPA CHUPS FLOWER BOUQUET, BONBON STICK
LAY'S	ELVIRA, SOSO, CORREZON, NIVEA, BLANCPECHE, SAUSSOUN

[a] La distance utilisée est une similarité cosinus sur les poids estimés dans le plongement de mot

- Des concepts appris, mais sur des produits de grande consommation, pour lesquels on a en général des échos

- Qualité des matchs ≈ Pipeline Elastic,
  - mais coûteux à intégrer à la pipeline avec d'autres critères de façon aussi flexible que ce que permet Elastic
  - parfois difficile d'expliquer la proximité entre libellés qui a été apprise

- Utilisation en complément pour repérer les faux positifs de la Pipeline Elastic
  - Ajuster la pipeline en fonction des erreurs repérées
  - Désaccords au sens de la distance siamoise repèrent les cas problématiques.

Taux d'appariements globaux

- Calories: 98% des produits enrichis (part supérieure en CA)
  - Matching `OpenFood` avec Elastic permet d'enrichir ≈ 40% de nos produits
  - Matching `CIQUAL` permet ensuite de rattraper ≈ 5% de nos produits

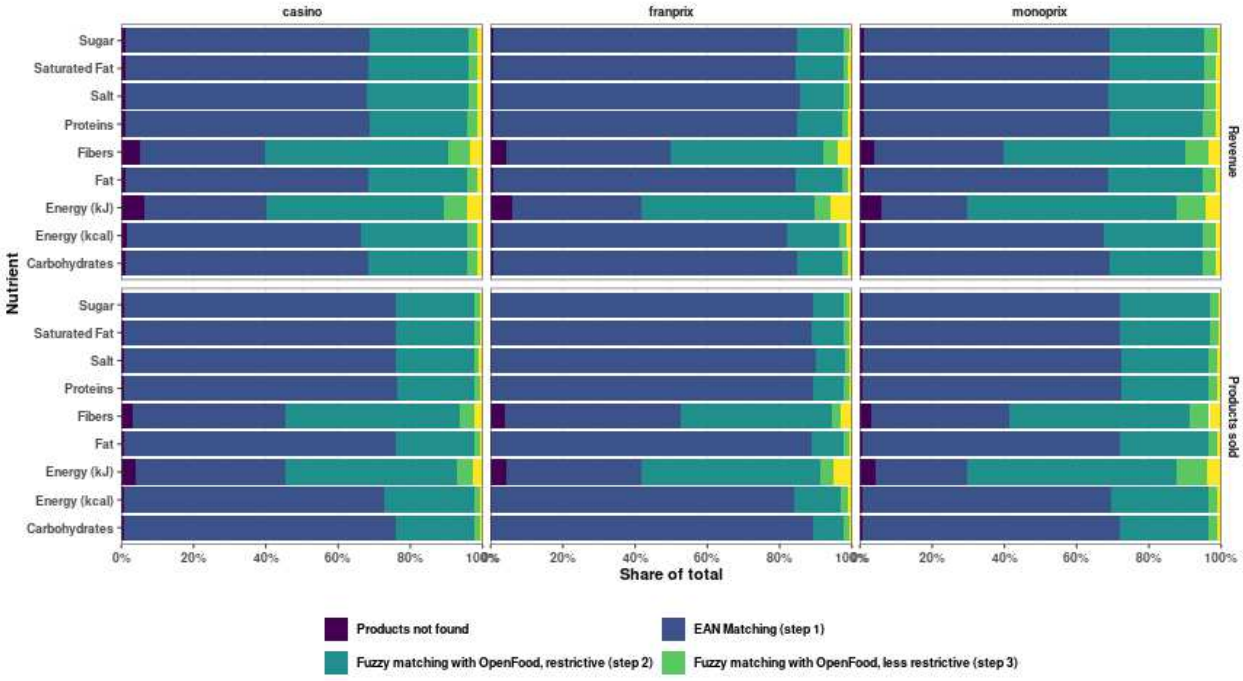
Random examples of the values linked between sources

RelevanC label		Open Food Facts label	Nutrients
Original Label	**Preprocessed label**	**Preprocessed label**	**Energy (by 100g)**
HERBE MENTHE POT	herbe menthe pot	menthe bio pot	180
MIEL ROMARIN HAUTE VALLES 480G	miel romarin haute valles	miel romarin	336
POTE AUVERGNATE 400G	pote auvergnate	truffade auvergnate	682
LE PANIER FAISSELLE BIO 4X100G	panier faisselle bio	panier	389
COCA-COLA ZERO PET 1.5LX6 CONT MAST	cocacola zero pet cont mast	cocacola pet	126
NAVARIN AGNEAU 1,2KE	navarin agneau	navarin petit legumes agneau francais	197
ABRICOT 35/45 BQ 1KG	abricot	abricot	84
BISTROT DE FRANCE RS BIB 5L	bistrot france rs bib	rs	1540
SAUMON SAUVAGE PROV.MSC 330G BQ	saumon sauvage prov msc	msc oeufs saumon sauvage	866
ENTREMETS CITRON MERINGUE 6P 500G	entremets citron meringue	cone citron meringue	1142

Taux d'appariements globaux (1/2)

- Performances proches pour toutes les catégories

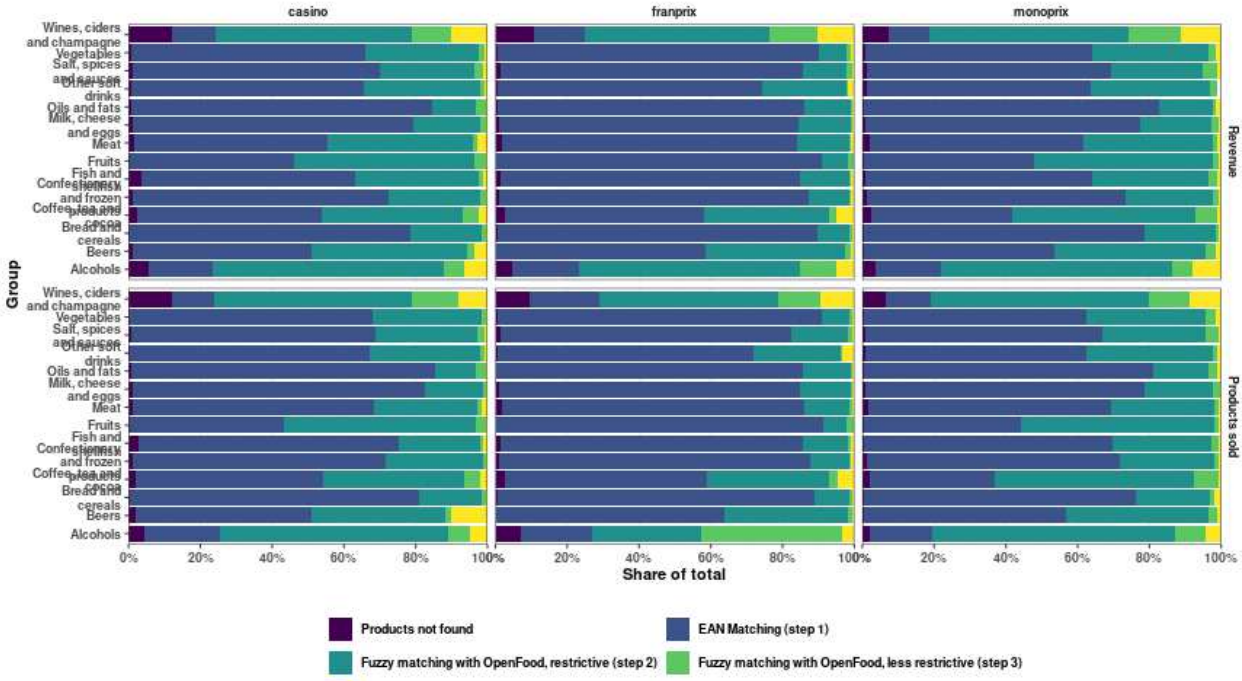
```
## [[1]]
```



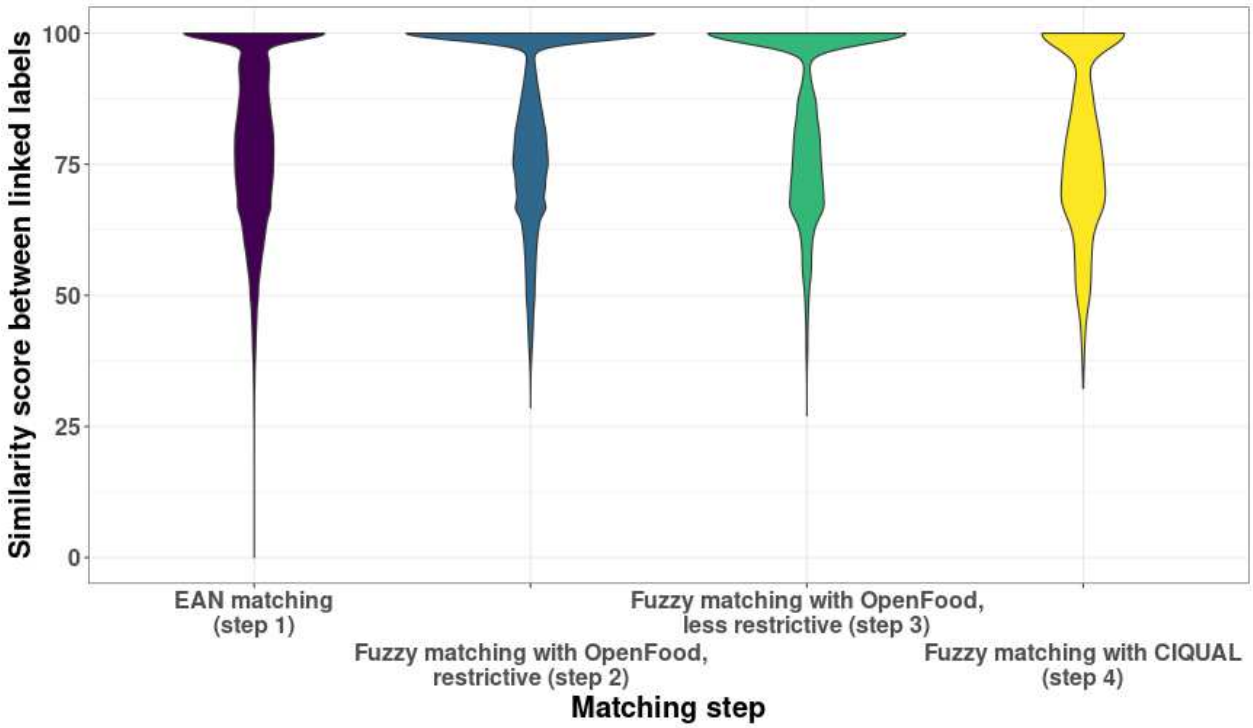
Taux d'appariements globaux (2/2)

- Des différences selon les groupes:

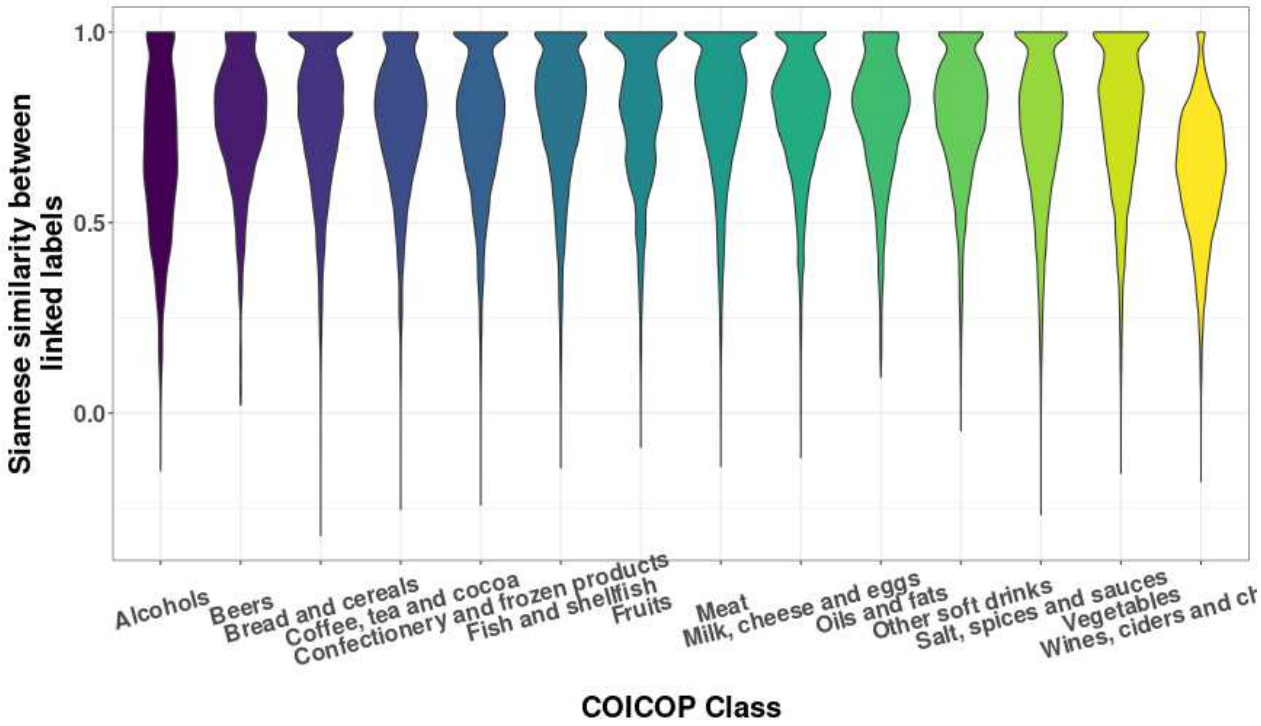
```
## [[1]]
```



Performance de l'appariement (1/2)



Performance de l'appariement (2/2)

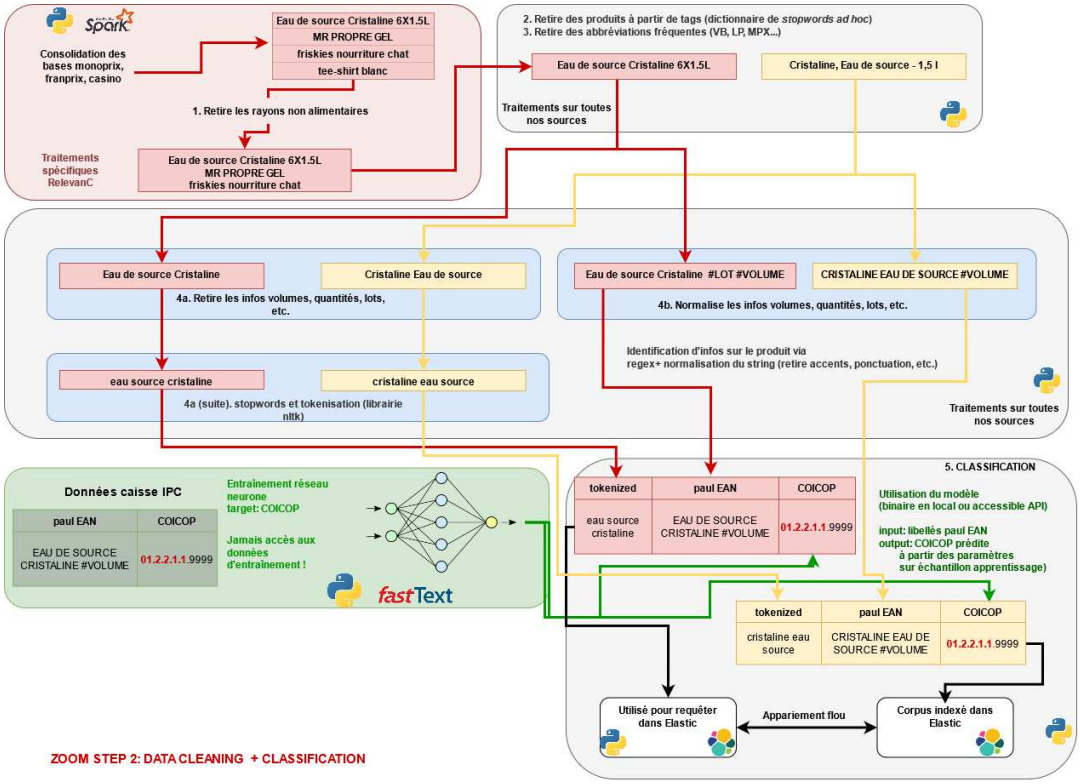


Appariements

- Appariements flous multiples:
  - 98% des magasins géolocalisés, 98% des produits enrichis grâce à l'`OpenFood` ou `CIQUAL`
  - Marges d'amélioration sur la validation *ex-post* des échos
- Gains énormes à utiliser la technologie `ElasticSearch` en l'absence d'identifiants exacts:
  - Recoder appariements flous en 🐍 ou Ⓡ purs serait une mauvaise idée
  - Prise en main minimale très rapide de l'outil (merci à ceux qui nous ont aidés ! 👍)
  - Encore des gains possibles

Perspectives

- *Pipeline* asynchrone du fait des données présentes dans des environnements différents
- Enjeu de la temporalité des données
    - `IRI`, `OpenFood` et les données de caisses sont des données qui évoluent
    - Privilégier l'API d'`OpenFood` pour la mise à jour ?
    - `ElasticSearch` permet de tenir compte de critères de temporalité dans les requêtes et l'indexation
- Outils existent pour améliorer l'entraînement des modèles et le déploiement de ceux-ci pour une mise à disposition:
    - *Un exemple*: `MLFlow` sur `SSP-cloud`

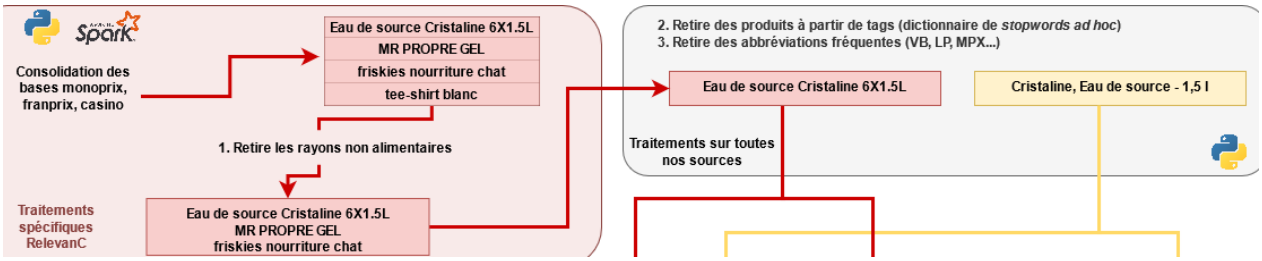


ZOOM STEP 2: DATA CLEANING + CLASSIFICATION

[Retour à la présentation]

Premières étapes: définir le champ le plus précis et cohérent possible

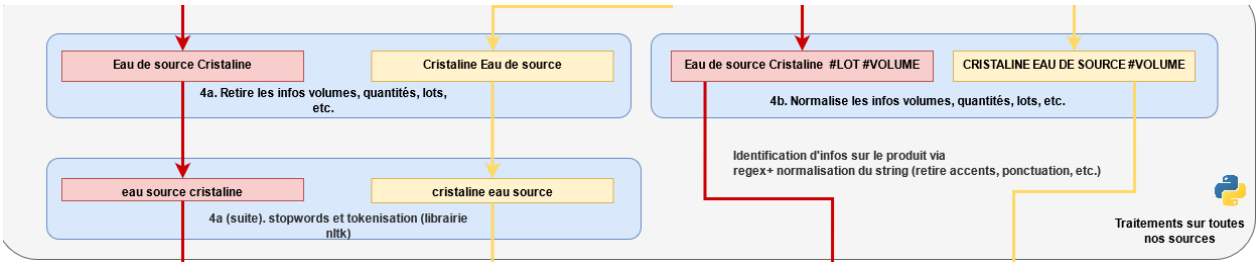
1. Consolidation des bases des différentes chaînes
2. Elimine des produits non alimentaires (`RelevanC` et `OpenFood`)
3. Retire des abbréviations qui apportent du bruit



Retour à la présentation

Premières étapes: définir le champ le plus précis et cohérent possible

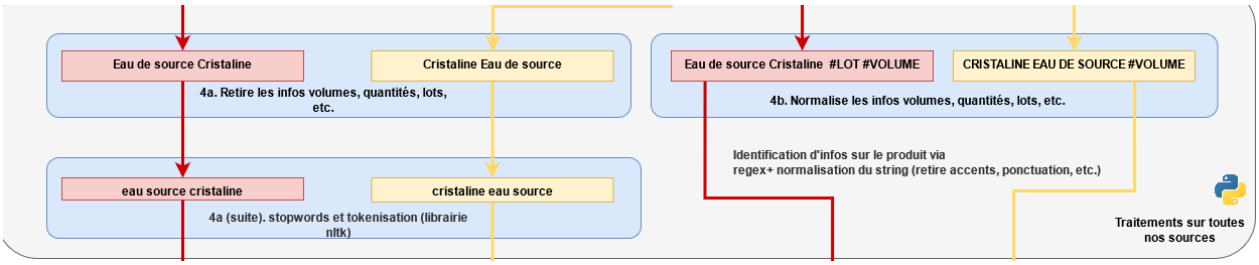
1. Identification par expression régulière d'infos sur le produit
    - Remplacement par des mots-clés (`#VOLUME`, `#POIDS`, `#UNITE`, `#LOT`...) pour la classification
    - Suppression de ces infos pour le libellé d'appariement flou



[Retour à la présentation]

Deuxième étape: normalisation des noms de produits

1. Transformation en suite d'unités signifiantes (`token`)
    - Retrait des *stopwords* qui n'apportent pas d'info sur le produit
    - Mots comme *token*
    - `nltk` plutôt que `spaCy`
    - Pas de `stemming` (défini au niveau de la requête `Elastic`)
    - Les *n-grams* sont définis au niveau de la requête `Elastic`



Retour à la présentation

Référentiels IRI

- Référentiel de **produits** et de **magasins** achetés par l'Insee
  - Mis à jour toutes les semaines
- Une base intéressante pour les **appariements exacts** :
  - Bon taux d'appariement EAN
  - Informations utiles: bio, poids du produit...
- Libellés propres:
  - $\neq$ données de caisses collectées automatiquement
- Exploite pour la **dimension poids**:
  - Nécessaire de creuser pour d'autres utilisations...
  - ... appariements flous notamment
- Référentiel des points de ventes IRI :
  - Enseignes
  - Coordonnées géographiques