

MA - Web Mining

Des recettes et des prix – Rapport



Lausanne, le 18.06.2023

Étudiants

Gugler Anthony

Hofmann Florian

Jordan Jérémy

<https://gitlab.forge.hefr.ch/florian.hofmann/webm-projet>

<https://github.com/Craypap/WebMining-Project>

But général de votre projet

Dans le cadre du cours Web Mining, nous réalisons un projet de groupe pour mettre en pratique les connaissances acquises durant le cours et les étendre. Nous allons récupérer des données de sites internet puis les analyser pour récupérer la valeur de celles-ci.

Le nom du projet est “*Des recettes et des prix*”.

Le projet se déroule du 21.04.2023 au 18.06.2023 avec une défense de ce projet le 23.06.2023.

1. Contexte et objectifs

Nous avons choisi de combiner le célèbre site de recette Marmiton avec deux autres sites pour récupérer le prix des ingrédients : le shop de Aldi et le site Union Suisse des paysans.

Ensuite, avec ces données, il sera possible de réaliser des analyses pour établir si l'estimation du coût des recettes par Marmiton est correcte. La corrélation entre l'estimation et le prix réel sera donc réalisée. Pour mettre en forme les résultats des analyses, un site web dynamique sera développé.

Les trois types de prix ci-dessous vont être présentés et comparés.

- Le prix du panier Aldi: ce prix prend en compte l'entier des produits achetés, par exemple : achat de 1 kg de carottes dans un paquet.
- Le prix à la quantité : ce prix prend en compte le grammage de la recette. Ainsi, ce prix est le minimum à payer pour réaliser la recette.
- Le prix moyen : le prix moyen entre Aldi et le prix indicatif chez Union des paysans.

De plus, les prix d'Aldi et les prix conseillés vont être comparés.

2. Etat de l'art

La mise en corrélation d'ingrédients de recettes de Marmiton et deux autres sites qui répertorient des prix pour une gamme de produits pour USP et des prix pour chaque produit pour Aldi est quelque chose d'innovateur. Aucun projet similaire n'a été trouvé sur internet.

Les techniques utilisées pour l'analyse des données ne sont par contre pas novatrices, des corrélations et des techniques de clustering sont utilisées.

3. Présentation des données

Ce chapitre présente les données ainsi que ses sources. Elles sont extraites à l'aide des méthodes de scraping conventionnel.

3.1 Marmiton

3.1.1 Général

Le site Marmiton <https://www.marmiton.org/> regroupe des recettes participatives commentées, notées et triées dans différentes catégories comme entrée ou dessert. Les ingrédients et une estimation de prix de la recette au global 1 (bon marché), 2 (moyen) et 3 (assez cher) sont présentés.

3.1.2 Droit d'utilisation

Dans les conditions générales d'utilisation¹, il est indiqué qu'il est seulement possible d'utiliser les données récupérées sur le site pour un usage privé. L'usage à but éducatif devrait aussi être accepté.

3.1.3 Description (attributs, quantité)

Nous récupérons les recettes avec 4 attributs qui leur correspondent. En plus, nous prenons une liste des ingrédients qui se caractérisent chacun de 3 autres attributs.

- nom
- catégorie (Entree, Plat principal, Dessert, Apéritif, Boisson)
- estimation de prix (1 pour *bon marché*, 2 pour *moyen* et 3 pour *assez cher*)
- nombre de portion de la recette
- liste des ingrédients contenant le nom et la quantité

¹ <https://www.marmiton.org/sp/aide/conditions-generales-utilisation.html>

```
{
  "name": "Flan de courgettes",
  "category": "Entree",
  "price_range": "1",
  "servings_quantity": "6",
  "ingredients": [{
    "name": "courgette",
    "quantity": "1 kg"
  }, {
    "name": "gruyère râpé",
    "quantity": "100 g"
  }]
  "url": "https://www.marmiton.org/recettes/Flanblabla.aspx",
},
```

Tableau 1 : Exemple d'une recette marmiton de base

3.1.4 Extraction (méthodes)

Nous avons utilisé un script python avec Scrapy. Les recettes sont choisies en parcourant les listes des recettes triées par catégories. Ainsi le scrapping se fait en partant de 5 urls (comme <https://www.marmiton.org/recettes/index/categorie/entree/>) qui vont parcourir toute leur liste de recettes et entrer dans chacune d'elles pour récupérer les détails intéressants.

3.2 Aldi

3.2.1 Général

Le shop de Adli <https://www.aldi-now.ch/> présente des articles disponibles à l'achat comme ceux qui pourraient être retrouvés dans les magasins physiques.

3.2.2 Droit d'utilisation

Dans les conditions générales et la politique de confidentialité fournies sur le site internet, il n'y a pas de mention explicite concernant la récupération des données pour la récupération de données de manière automatique (scraping). Cependant, la plupart des entreprises considèrent le scraping de données comme une violation de leurs conditions d'utilisation, et cela peut entraîner des conséquences juridiques si cela est découvert.

Toutefois, pour une utilisation dans le cadre privé ou à but éducatif, cela est généralement accepté.

3.2.3 Description (attributs, quantité)

Nous récupérons les 4 éléments ci-dessous pour tous les articles du site pour réaliser les différentes analyses présentées.. Nous récupérons tous les articles, il y en a environ 2'900 articles.

- nom
- poids
- prix de référence (au 100g par exemple)
- prix

```
{  
  "name": "BBQ Petits pains, fromage",  
  "weight": "250 g",  
  "price_per_unit": "(0.80/100 g)",  
  "price": "1.99"  
},
```

Tableau 2 : Exemple d'un produit Aldi de base

3.2.4 Extraction (méthodes)

Nous avons utilisé un script python avec Scrapy. Nous arrivons sur la page <https://www.aldi-now.ch/fr/nouveau-dans-la-boutique-en-ligne?ipp=72> ,, nous récupérons les informations de cette page (72 articles) puis nous passons aux pages suivantes. Ensuite, nous passons aux prochaines catégories avec la même méthodologie.

3.3 Union Suisse des paysans

3.3.1 Général

Le site Union Suisse des paysans <https://www.sbv-usp.ch/fr/prix/vente-directe> regroupe les prix indicatifs à titre informatif pour les producteurs pour la vente. Le site en lui-même ne permet pas d'acheter de produits.

3.3.2 Droit d'utilisation

Comme pour le shop de Aldi, il n'y pas de mention explicite sur le site web. L'utilisation à but éducatif devrait être acceptée.

3.3.3 Description (attributs, quantité)

Nous récupérons les 3 éléments ci-dessous pour tous les produits que liste le site. Il y a nettement moins de produits que Aldi (environ 200 produits), mais ce site vient en backup de Aldi pour des ingrédients "de base".

- nom
- quantité (pour le prix indiqué)
- prix

```
{  
  "name": "Fraises, en barquette",  
  "quantity": "kg",  
  "price": "13.00 - 18.00"  
}
```

Tableau 3 : Exemple d'un produit USP de base

3.3.4 Extraction (méthodes)

Nous avons utilisé un script python avec Scrapy. Les 8 pages qui listent différentes catégories de produit sont scrapées les unes après les autres pour en tirer les éléments cités au point ii.3 ci-dessus.

4. Prétraitement

Ce chapitre présente le prétraitement effectué sur les données des trois sources. Ces étapes de nettoyage ont permis de rendre les données prêtes et plus faciles pour analyser.

4.1 Aldi

Après avoir récupéré les données brutes selon la méthode présentée précédemment dans ce rapport, plusieurs étapes de nettoyage ont été effectuées. Tout d'abord, tous les doublons ont été supprimés afin d'éviter toute redondance dans les données. Ensuite, tous les prix à la quantité ont été uniformisés pour afficher le prix au kilogramme, arrondi à deux chiffres après la virgule. Finalement, les données sont stockées dans un même format entre Aldi et USP.

Ces tâches permettent d'avoir des informations à afficher avec un même format JSON et ainsi d'avoir le prix au kilogramme pour comparer les prix entre les différents ingrédients et le prix du produit.

```
{
  "source": "ALDI",
  "name": "ALPINA Saucisses au fromage et au piment",
  "price_kg": 13.4,
  "price": 6.99,
  "ingredient": "",
  "link": "https://www.aldi-now.ch/fr/alpina-saucisses-au-fromage- et-au-piment/8862890131457"
}
```

Tableau 4 : Exemple d'un produit Aldi après le prétraitement

4.2 USP

Pour les données du site de l'union suisse des paysans (USP), le même traitement que les données Aldi est effectué. De ce fait, toutes nos données "ingrédients" sont transformées dans le même format. On garde malgré tout une trace de la source de la donnée avec un champ `source` qui indique le site source, donc USP ou Aldi.

```
{
  "source": "USP",
  "name": "Haricots secs",
  "price": -1,
  "ingredient": "",
  "price_kg": 80.0
}
```

Tableau 5 : Exemple d'un produit USP après le prétraitement

4.3 Marmiton

Le prétraitement de Marmiton se fait après celui d'Aldi et USP. Il a pour but d'associer un ingrédient présent dans les recettes à un produit USP ou ALDI et d'ajouter une catégorisation des prix aux recettes selon les prix des produits. Il en ressortira 2 fichiers qui seront les deux seuls qui seront indexés dans elasticsearch.

En premier lieu, les ingrédients des recettes de Marmiton sont retirés et triés pour éliminer les doublons. La liste distincte est ensuite écrite dans un JSON pour but de comparatif.

Ensuite, les produits de Aldi et USP sont mis dans une même liste et parcourus pour y associer un ingrédient. La manière de faire est de voir de manière stricte si la donnée correspond à un ingrédient. Si la première recherche est infructueuse, plusieurs autres vont suivre en raccourcissant le nom du produit, puis celui des ingrédients. Ceci met en place une affiliation optimale vu qu'il la sévérité est réduite avec le temps.

Étant donné que les ingrédients de Marmiton manquent d'homogénéité (parfois l'ingrédient "pomme" est mis comme "pomme épluchée" ou "pomme bien mûre, miam") ce qui fait que la majorité n'est pas affiliée. C'est pour ça que la nouvelle entrée "other_ingredients" pour les produits est ajoutée. Elle montre l'affiliation produite à d'autres ingrédients potentiellement similaires.


```
{
  "source": "USP",
  "name": "Framboises",
  "price": -1,
  "ingredient": "framboises",
  "price_kg": 23.0,
  "other_ingredients": ["banane en rondelles ou une poignée de framboises", "citron et/ou orange non traité...", "baies rouge comme des framboises",...]
},
```

Tableau 6 : Exemple de recette après le prétraitement

Ensuite, un dernier traitement est fait sur les données des recettes. Il est question d'affilier deux nouvelles catégories de prix (selon ALDI et USP) en utilisant un clustering. Pour se faire, les recettes sont parcourues et on affine un prix sur ses ingrédients selon le premier résultat de la recherche des produits affiliés. Le clustering regroupe ensuite en 3 catégories pour donner les données finales des recettes :

```
{
  "name": "Flan de courgettes",
  "category": "Entree",
  "price_range": "1",
  "servings_quantity": "6",
  "ingredients": [{
    "name": "courgette",
    "quantity": "1 kg"
  }, {
    "name": "gruyère râpé",
    "quantity": "100 g"
  }]
  "url": "https://www.marmiton.org/recettes/Flanblabla.aspx",
  "ALDI_cluster": 1,
  "UPS_cluster": 0
},
```

Tableau 7 : Exemple de recette finale

4.4 Calcul des prix et clustering

Ensuite, le prix de chaque recette est calculé pour les deux sources de données (USP et Aldi) dans la continuité du prétraitement présenté dans ce chapitre. Ces prix calculés permettent de réaliser des regroupements en fonction du prix (clustering).

5. Conception / cas d'utilisation / Architecture

5.1 Conception

La Figure 1 présente le diagramme de comment les données arrivent jusqu'à l'indexation des données sur Elastic.

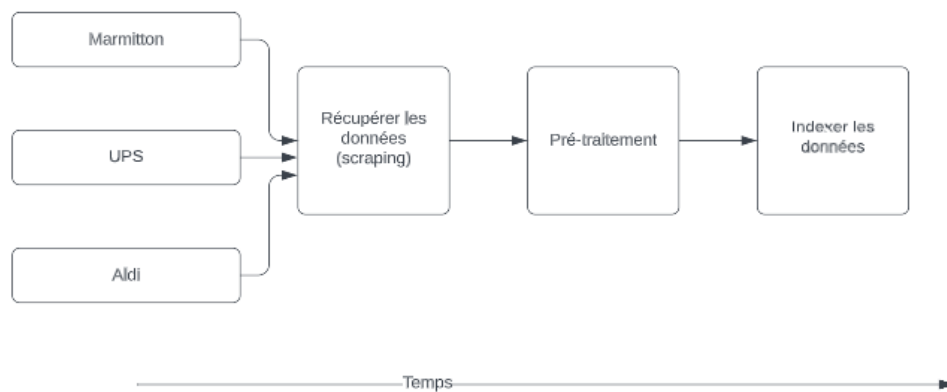


Figure 1 : Diagramme présentant comment les données sont récupérées, transformées et indexées

La Figure 2 présente le diagramme de séquence de l'application. L'utilisateur recherche une recette sur l'application. Les ingrédients et les informations sont affichés si ceux-ci existent pour la recette recherchée.

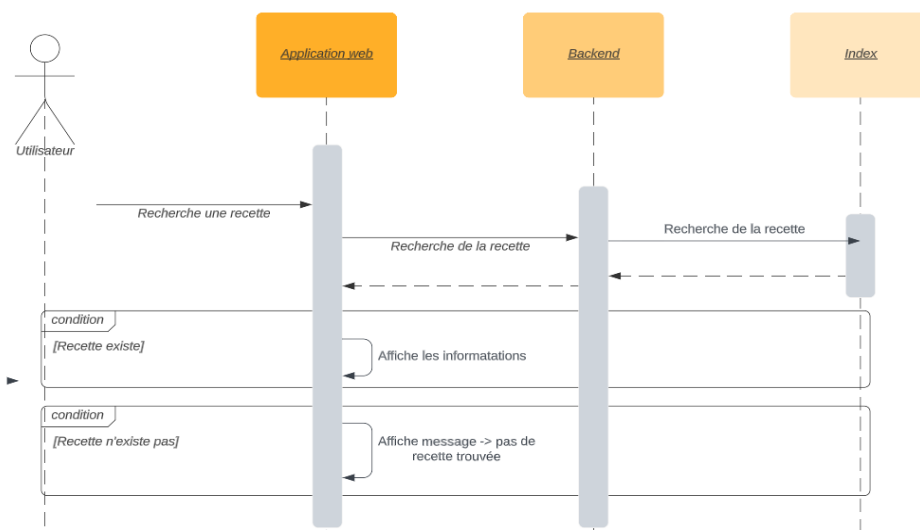


Figure 2 : diagramme de séquence

5.2 Cas d'utilisation

La Figure 3, ci-dessous, présente le cas d'utilisation de l'application du projet. Il y a donc trois grandes fonctionnalités. L'utilisateur peut rechercher une recette, ce qui permet d'afficher les informations (ingrédient, prix des deux sources) si ceux-ci existent. De plus, l'utilisateur peut mettre à jour les données en cliquant sur un bouton. Ceci va récupérer les données, les prétraiter et les indexer.

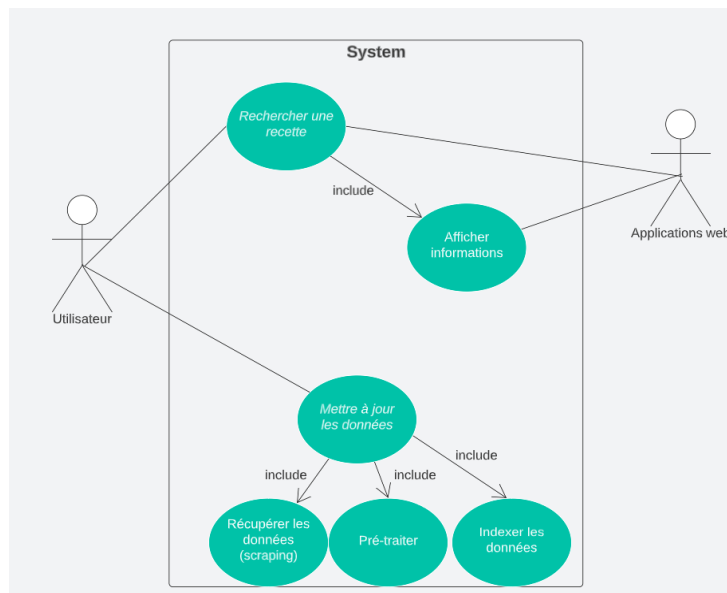


Figure 3 : diagramme de cas d'utilisation

5.3 Architecture globale / technologie envisagée

L'architecture envisagée est une architecture classique “Client - serveur - base de données”, illustrée la Figure 4.

Le client serait une page WEB réalisée en *Angular*², le serveur (qui contiendra le scraper, analyser, ...) sera en python avec le framework WEB *FastAPI*³. La base de données sera *ElasticSearch*⁴ pour indexer et structurer nos données scrapées

² <https://angular.io/>

³ <https://fastapi.tiangolo.com/>

⁴ <https://www.elastic.co/fr/elasticsearch/>

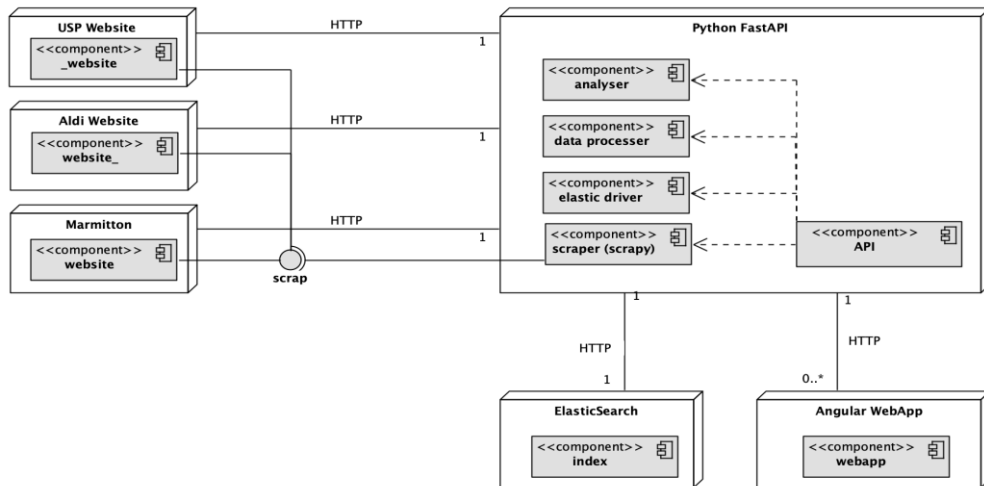


Figure 4 : Architecture

6. Fonctionnalités

6.1 Rechercher une recette

Sur l'application web, l'utilisateur peut rechercher une recette de Marmittion. Pour cela, il suffit de taper le nom d'une recette dans la barre de recherche qui se trouve sur la page d'accueil. La Figure 5 ci-dessous illustre la page d'accueil de l'application.



Figure 5 : Page d'accueil

Un message d'erreur est affiché si la recette recherchée n'existe pas ou n'est pas trouvée. Cette action est illustrée par la Figure 6.

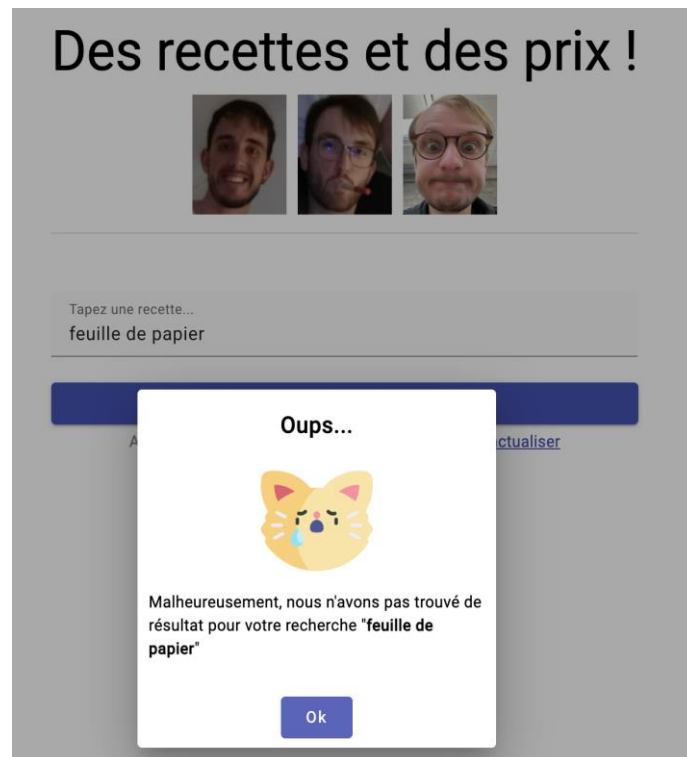
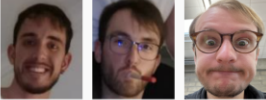



Figure 6 : Message d'erreur

6.2 Afficher les informations

Les informations, à savoir les ingrédients et les divers prix, sont affichées si la recette existe dans la base de données. Un classement des recettes selon 3 catégories (1, 2 ou 3 \$) est aussi fait et affiché. Il prend comme base les prix des ingrédients venant d'ALDI et séparés selon un clustering expliqué dans le chapitre suivant. De plus, l'utilisateur peut modifier la quantité prévue par marmiton et voir en direct l'évolution du prix qu'il va devoir déboursier. Toutes ces actions sont illustrées par la Figure 7 ci-dessous, qui est une capture d'écran de la vue après avoir trouvé la recette recherchée.



Voici le résultat de votre recherche :



Fondue d'endives [4 pers. | pce.] \$\$\$

Plat principal

Recette complète [ici](#) !

Liste des ingrédients à acheter*

Ingrédient	Quantité	Article proposé	Pièce(s)	Prix**	
beurre	40 g	Le Beurre (modifier)	x1	3.95	
crème fraîche	30 cl	MILFINA Crème fraîche 35% (modifier)	x2	5.58	
sucre	1 cuillère	Sucre cristallisé fin (modifier)	x1	1.29	
poivre au moulin		JUST VEG! Filet au poivre végétalien (modifier)	x1	2.09	
sel		JURASEL Sel (modifier)	x1	1.05	
endives	4	Endives suisses 500 g, SUISSE GARANTIE (modifier)	x2	4.58	
citron		BELLA Sucre aromatisé, citron (modifier)	x1	1.99	

Prix à la caisse : 20.53 CHF

* Les recettes sont tirées de Marmiton.fr

** Les prix sont basés sur Aldi.ch

Nouvelle recherche

Projet "Web Mining", MSE 2023

Figure 7 : Vue des ingrédients d'une recette

Si un ingrédient proposé ne correspond pas tout à fait, l'utilisateur a toujours la possibilité de changer parmi une sélection de produit qui sont potentiellement les bons. Pour ce faire, il appuie sur le texte "modifier" à côté de l'article proposé, et un pop-up s'ouvre pour lui proposer une sélection de produit, comme l'illustre la Figure 8.

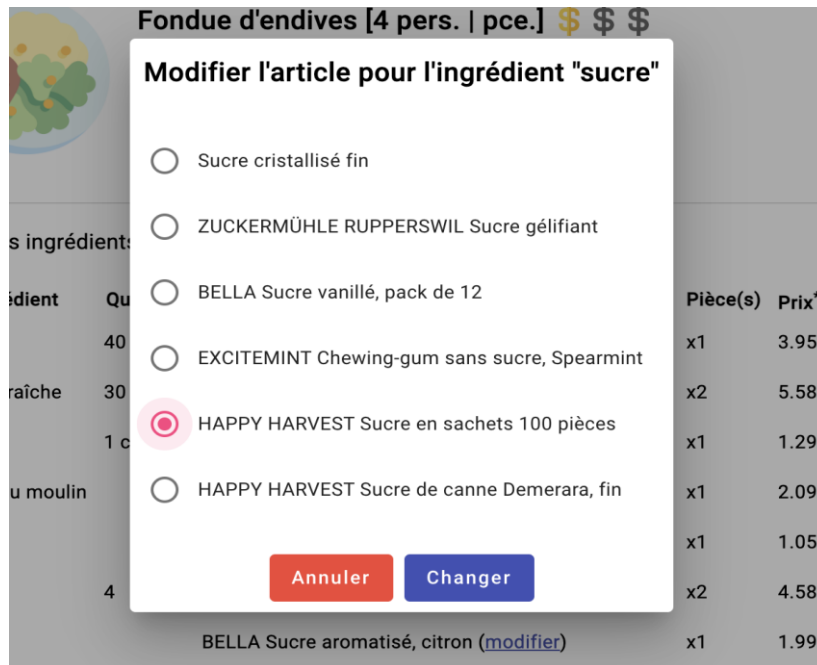


Figure 8 : Pop-up de sélection d'articles similaires

6.3 Mettre à jour les données

L'utilisateur a la possibilité de mettre à jour les données, ainsi avoir les dernières recettes et prix. Les informations sont récupérées des trois sources, puis prétraitées et finalement indexées pour pouvoir afficher les données récentes. Pour ce faire, il doit appuyer sur le texte "actualisé", à côté d'indication de la dernière actualisation, sur la page d'accueil, comme l'illustre la Figure 9 ci-dessous.



Figure 9 : Actualisation des données

7. Techniques, algorithmes et outils utilisés

Pour la reconnaissance des produits selon les ingrédients, la technique est faite en prétraitement est expliquée plus en détail sous le chapitre 4.3. Aucun algorithme précis n'est utilisé autre que la reconnaissance de mots dans une suite d'autres mots.

Concernant le regroupement de recettes selon 3 catégories de prix, il fallait en premier lieu donner un prix aux ingrédients des recettes et de les additionner. Il a été fait en prenant le premier produit trouvé qui correspondait à l'ingrédient et d'adapter son prix selon la quantité. Ensuite, après avoir calculé le prix de chaque recette, il fallait les regrouper. Il avait été défini dans le cahier des charges de faire du clustering. Le choix de l'algorithme s'est posé sur K-Means plutôt que DBSCAN. Les deux sont fonctionnels, mais K-Means était plus rapide. La tâche se fait en quelques secondes, ce qui s'explique par la quantité de données et sa dimension unique. Le cluster se fait une fois par sources de prix (ALDI puis USP) permettant de faire la distinction. Voici les résultats :

ALDI	USP
{	{
0: {	0: {
'range': [0.0, 10.64100],	'range': [-2.585, 10.705],
'count': 23873	'count': 26277
},	},
1: {	1: {
'range': [10.6429, 23.595],	'range': [10.715, 43.44],
'count': 13227	'count': 12540
},	},
2: {	2: {
'range': [23.6289, 133.269],	'range': [43.5250, 131.6250],
'count': 2029	'count': 312
}	}
}	}

Tableau 8 : Détails du clustering de Aldi et USP

Il est important de noter que le clustering d' UPS ne va pas être utilisé. En effet, le peu de produits correctement orthographiés ne permet pas de juger de manière pertinente la majorité des recettes. Si un ingrédient manque, le prix de la recette ne change pas. En plus, étant donné que certaines quantités sont manquantes (si l'ingrédient est 1 pomme, alors il n'y a pas de quantité en kg) alors c'est le champ "price" qui est utilisé ce qui correspond à "-1". C'est pourquoi certaines recettes ont des prix négatifs.

8. Planification, organisation et suivi répartition du travail (diagramme de Gantt)

Voici ci-dessous le planning pour le projet.

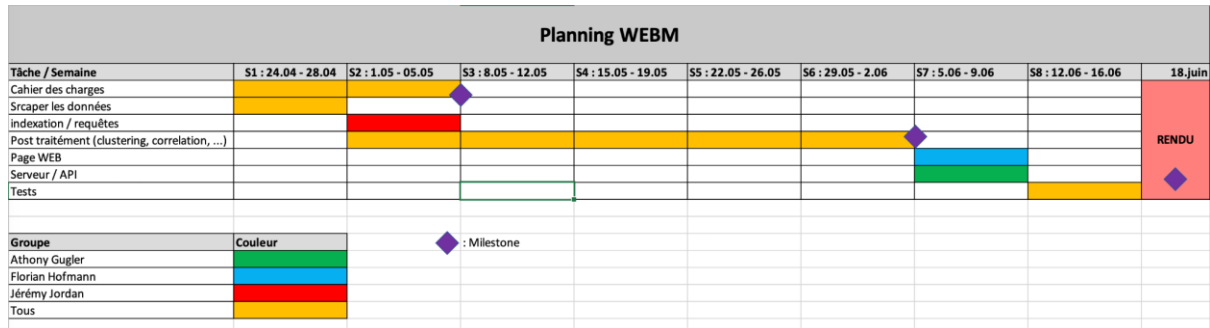


Figure 10 : Planning

Milestones :

Voici ci-dessous les jalons pour les rendus à faire durant le projet.

- M1 (07.05.2023) : Cahier des charges + scraping
- M2 (04.06.2023) : Analyse des données
- M3 (18.06.2023) : Application finale + rapport

9. Résultats

9.1 Analyse des résultats

Les résultats des prix des recettes sur l'application web ne peuvent pas directement être de confiance. Il n'est pas rare de voir une mauvaise affiliation des produits à des ingrédients, mais il est possible de changer dans l'application donc ce n'est pas si grave. En plus, certains prix ont dû être arrondis/supposés par manque d'information sur la quantité, ce qui impacte l'exactitude des résultats, surtout quand celui-ci n'avait pas d'unité (1 pomme vs 1 pastèque). Néanmoins, après quelques ajustements de la part de l'utilisateur, le prix est généralement assez représentatif de la réalité.

L'énorme défi de ce projet est l'affiliation d'un ingrédient à un produit. Il requiert une énorme analyse et un traitement des noms des produits et des ingrédients ce qui impacte forcément le résultat final. Notre méthode d'affiliation décrite dans le chapitre de prétraitement de marmiton affilié principalement 377 ingrédients sur les 32'000 différents (donc environ 1%) à 2649 produits sur les 3'100 différents (donc environ 85%). Ceci relève que beaucoup d'ingrédients distincts ne sont pas clairs. Il est dur de savoir quel produit affilié à l'ingrédient *"vanille bien sûr, mais pourquoi pas châtaigne, caramel, crème brûlée... voire même, poire, figue, banane, tiramisu... Soyons fous !"*. C'est pour ça qu'il fallait mettre le deuxième champ "other_ingredient" qui était bien plus laxiste pour affilier des ingrédients secondaires.

9.2 Résultats "Visuel"

Le résultat visible par l'utilisateur final, donc principalement l'application WEB, correspond à ce qui a été pensé lors de l'élaboration du cahier des charges. Même si aucunes maquettes (ou autres outils de conception UI / UX) ont été utilisées, le résultat est celui attendu. Cependant, nous avons décidé d'amputer quelques fonctionnalités prévues, notamment d'afficher différents types de prix au profit du prix que paierait le consommateur s'il achetait tous les articles nécessaires au magasin Aldi. Nous nous sommes dit qu'il valait mieux faire une application avec un seul but, dans notre cas présenter les articles disponibles chez Aldi à acheter pour une recette, plutôt que d'avoir trop de fonctionnalités différentes qui pourraient rendre l'utilisateur confus. Nous avons tout de même écrit du code python (`price_analysis.py`) qui imprime en console une analyse comme imaginée initialement afin de pouvoir quand même faire une comparaison, sans surcharger l'application web d'informations.

9.3 Analyse avec le fichier Price_analysis.py

Ce fichier⁵ fournit une interface visuelle dans la console pour comparer les prix des ingrédients d'une recette à partir de différentes sources, comme illustré à la *Figure 11*. Il faut simplement mettre l'index de la recette présent dans Elastic dans le main de ce fichier. Chaque ingrédient est affiché avec son prix unitaire, son prix au kilogramme, et son coût à la quantité pour la recette (partie de gauche de la *Figure 11*). À la fin de la sortie, un total des coûts est affiché pour les deux sources et pour les divers prix mentionnés dans ce chapitre, ce qui facilite la comparaison des prix (partie de droite de la *Figure 11*).

En évaluant les résultats, nous avons constaté qu'USP ne fournit pas suffisamment de produits pour une comparaison complète des coûts de recette entre deux sources. Il serait peut-être plus sensé de comparer uniquement les prix des produits disponibles chez USP.

De manière générale, Aldi, étant un discounter, propose des prix qui sont généralement similaires aux prix recommandés par USP.

```
Profiteroles express
Source : ALDI
Ingrédient original : chocolat noir
Ingrédient : chocolat
Prix par kg : 18.6
Prix direct : 1.49
Prix selon quantité : 1.8600000000000003
-----
Source : USP
Ingrédient original : beurre
Ingrédient : beurre
Prix par kg : 15.5
Prix direct : -1
Prix selon quantité : 1.55

Total cost for USP ingredients:
Price per kg: 83.25
Direct price: -4
Price according to quantity: 8.32
-----
Total cost for ALDI ingredients:
Price per kg: 49.65
Direct price: 15.64
Price according to quantity: 7.55
-----
```

Figure 11 : Résultat dans la console pour la recette de Profiteroles express

⁵https://gitlab.forge.hefr.ch/florian.hofmann/webm-projet/-/blob/main/code/analyser/Price_analysis.py

10. Conclusion / travail futur

La conclusion de ce projet est positive. L'objectif principal qui est d'avoir un mini-moteur de recherche afin d'afficher le prix d'une recette est atteint. On se rend alors compte que les résultats ne sont pas toujours très juste, mais des mécanismes de correction ont été mis en place afin de pallier ce défaut. Hormis ce détail, l'application est fonctionnelle et nous sommes satisfaits du résultat.

Ce travail démontre la difficulté de retrouver de bons ingrédients chez un commerçant qui les nomme selon la marque du produit. Nous pensons également que ce genre d'application mériterait d'être développée plus minutieusement afin d'avoir des résultats encore meilleurs. Cela permettrait à un bon nombre de personnes de se rendre compte du coût d'une recette. En allant plus loin, il serait possible de vérifier l'impact du prix sur l'augmentation de la quantité de portions. Ceci permettrait potentiellement d'éviter le gâchis alimentaire en faisant une plus grande portion pour manger un autre jour, et alors utiliser le plus possible des articles achetés.

Finalement, ce fut un projet très enrichissant pour nous. Cela a permis de mettre en pratique des techniques de scraping vues en cours dans un cas pratique que l'on pourrait alors rencontrer dans notre future vie professionnelle. Mais il serait aussi possible d'utiliser des notions de traitement de données vues dans d'autres cours auparavant. Tout ça afin d'aboutir à un résultat plus convaincant selon nous.

Comme travail futur pour améliorer la solution, nous voyons principalement les points suivants.

- Le point le plus important sera d'améliorer grandement le matching des ingrédients. Cela pourrait se faire en utilisant d'autres algorithmes, mais aussi en incluant d'autres commerces afin d'être plus sûr de trouver l'ingrédient recherché.
- Améliorer la recherche en permettant directement de mettre une URL marmiton, et aussi de proposer plusieurs positions comme le fait Google ou tout autres moteurs de recherche.
- Donner la possibilité à l'utilisateur d'indiquer les produits qu'il possède déjà chez lui afin de ne pas les compter dans le total à payer. Cela afin d'éviter aux utilisateurs de faire des mathématiques.