

Wine Recommender

Groupe D L. Hirschi, B. Pasquier, A. Petrucci

Ce repository GitHub contient tout le code de notre application ainsi que la documentation.

L'application Web finale est accessible à l'adresse suivante: webapp-wine-recommender.kube.isc.heia-fr.ch

Contexte et objectifs

Dans le cadre du cours de Web Mining, il est demandé de réaliser un projet de récupération et d'analyse de données. Le sujet choisi pour ce dernier est la création d'un système de recommandation de vin en fonction de recette.

Lorsque nous choisissons de préparer une recette, nous nous demandons parfois quel vin accompagnerait le mieux ce plat. En règle générale, les descriptifs de vins viennent accompagnés d'une recommandation du style "bœuf", "poulet", "poisson", etc. Mais ces recommandations sont souvent trop générales et ne permettent pas de choisir un vin adapté à la recette. De plus, il est difficile de trouver des informations sur les vins, notamment sur les vins de qualité moyenne ou inférieure. C'est pourquoi nous avons décidé de créer un système de recommandation de vin en fonction d'une recette sous forme d'une application web.

Données

Pour réaliser ce projet, deux sources de données sont utilisées :

- **Vivino** pour la récupération des données des vins.
- **Marmittton** pour la récupération des données sur les recettes.

Afin de traiter une quantité de données raisonnable, nous nous limitons aux plats principaux pour les recettes et aux vins rouges et blancs pour les vins. Il y a 1'843 vins (1'257 rouges et 586 blancs) et il y a 3'398 recettes. De base, il y avait ~17'000 recettes. En revanche, comme le temps de calcul afin de créer les embeddings pour faire la recommandation était trop longtemps, nous avons décidé de prendre uniquement 1 recette sur 5.

Nous effectuons différents prétraitements sur les données collectées. Lors de la récupération des données sur les sites que nous avons choisis, certaines opérations doivent être effectuées pour enlever certains caractères (comme les parenthèses ou certaines balises). Il a aussi fallu enlever tous les mots inutiles (de, des, la, le, ...) car nous ne les voulons pas pour créer l'embedding des similarités. Pour les recettes, nous enlevons aussi tous les mots qui apparaissaient moins de 5 fois dans toutes les recettes prises en compte, parce qu'il n'y a d'intérêt à avoir ces mots pour l'embedding.

Voici les données des recettes ainsi que leurs descriptions :

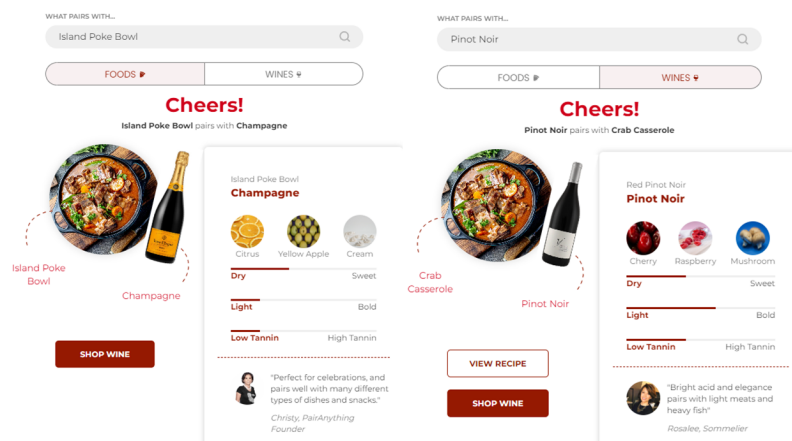
- **name**: le nom de la recette
- **type**: le type de plat (dans notre cas, il y a que des plats principaux)
- **ingredients**: la liste des ingrédients de la recette
- **url**: l'URL de la recette sur Marmiton
- **pairings_embedding**: le tableau contenant les valeurs de l'embedding afin de trouver la meilleure recette avec les meilleurs vins

Voici les données des vins ainsi que leurs descriptions :

- **name**: le nom du vin
- **winery**: la cave du vin
- **vintage**: l'année du vin
- **image_url**: l'URL de l'image du vin
- **country**: le pays du vin
- **region**: la région du vin
- **type**: le type de vin (dans notre cas, vin rouge ou vin blanc)
- **grapes**: la liste des cépages du vin
- **price**: le prix du vin
- **pairings**: la liste des ingrédients qui se marie bien avec ce vin
- **reviews**: la liste des avis du vin (les 3 avis jugés utiles par Vivino)
- **average_rating**: la moyenne des notes du vin
- **url**: l'url du vin
- **reviews_sentiment**: une liste avec le sentiment (négatif, neutre et positif) et sa valeur
- **reviews_emotion**: une liste avec les sentiments (autres, la joie, la tristesse, la colère, la surprise, le dégoût et la peur) et sa valeur
- **pairings_embedding**: le tableau contenant les valeurs de l'embedding afin de trouver la meilleure recette avec les meilleurs vins

État de l'art

Il n'existe pas beaucoup de projets ou de recherches disponibles en ligne qui traitent directement de la recommandation de vins à partir de recette. L'application Web [PairAnything](#) est le seul exemple que nous ayons pu trouver et qui est très semblable à l'application que nous voulons développer dans ce projet. En effet, cette application propose à l'utilisateur de choisir une recette et de proposer une sorte de vin qui s'accorde bien avec celle-ci. Elle permet également de faire l'opération inverse, c'est-à-dire trouver une recette à partir d'une sorte de vin. L'interface et les résultats de ces deux fonctionnalités sont décrits par l'image suivante.



Bien que cette application soit similaire à ce que nous voulons faire, nous ne connaissons pas son fonctionnement interne lui permettant de proposer un vin à partir d'une recette.

Il existe néanmoins d'autres travaux qui se focalisent sur la recommandation de vin à partir d'une description ou d'un vin similaire. Dans [cet article](#), les auteurs récupèrent 180'000 reviews de vins récupérés sur le site [Web Wine Enthusiast](#) et les traitent pour récupérer uniquement les mots décrivant les arômes des vins. Ils

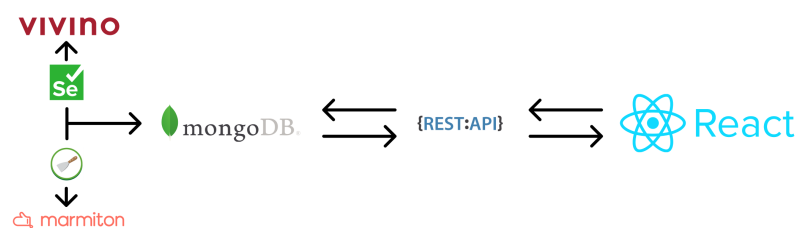
génèrent ensuite pour chacun d'eux un embedding à l'aide d'un modèle Word2Vec entraînés sur les 180'000 reviews. Ces embeddings leur permettent ensuite de trouver les vins similaires à un autre vin donné en entrée, et donc de les recommander à l'utilisateur.

Un [autre projet](#) similaire vise à recommander des vins à partir d'un texte donné par l'utilisateur. Des embeddings sont également générés à partir de 130'000 reviews. Cependant, la technologie [sentence-BERT](#) est utilisée et permet de générer ces embeddings à partir de phrases.

Il existe d'autres travaux visant à réaliser un recommandeur de vins à partir de données textuelles et la plupart d'entre eux se basent sur la génération d'embedding. Nous utilisons donc également cette technique afin de mapper les informations d'une recette (titre et ingrédients) aux accords mets-vins des vins.

Conception

L'architecture globale du projet est décrite par le diagramme suivant :



Données

Comme expliqué précédemment, les données sont récupérées sur les sites de Vivino et Marmiton à l'aide de Selenium et Scrapy respectivement. Ces dernières sont ensuite insérées dans une base de données.

Base de données

Le SGBD choisi pour ce projet est MongoDB, car c'est un système qui prend des "documents", c'est-à-dire des JSON, et les données récoltées sont également dans ce format. Il y a d'autres SGBD qui sont similaires, mais, par préférence, nous avons choisi celui-ci.

Afin d'y insérer les données, une interface MongoExpress permet d'afficher les tables dans un navigateur Web et d'interagir avec ces dernières.

API REST

Pour connecter notre backend et notre "frontend, une API Node est présente. On peut retrouver les différentes routes accessibles et les données qu'elles retournent sur le [README](#) dédié.

On utilise la librairie Javascript Mongoose, dédiée à la connexion aux bases de données de type MongoDB.

React

La dernière partie de l'architecture est le frontend qui permet à l'utilisateur d'interagir avec notre recommandeur de vins. Ce dernier est développé en React avec une interface Material UI.

CI/CD

Enfin, pour faciliter la mise en production de cette architecture, on utilise une pipeline GitHub Actions. Cette dernière est composée de deux parties: la première pour déployer le backend ainsi que la DB et la deuxième pour déployer le frontend.

Les dossiers *backend* et *frontend* du repository déclenchent respectivement leur workflow. Pour le backend, le déclenchement va d'abord build l'app Node et ensuite créer une image Docker avec les 3 composants :

- MongoDB
- MongoExpress
- Node

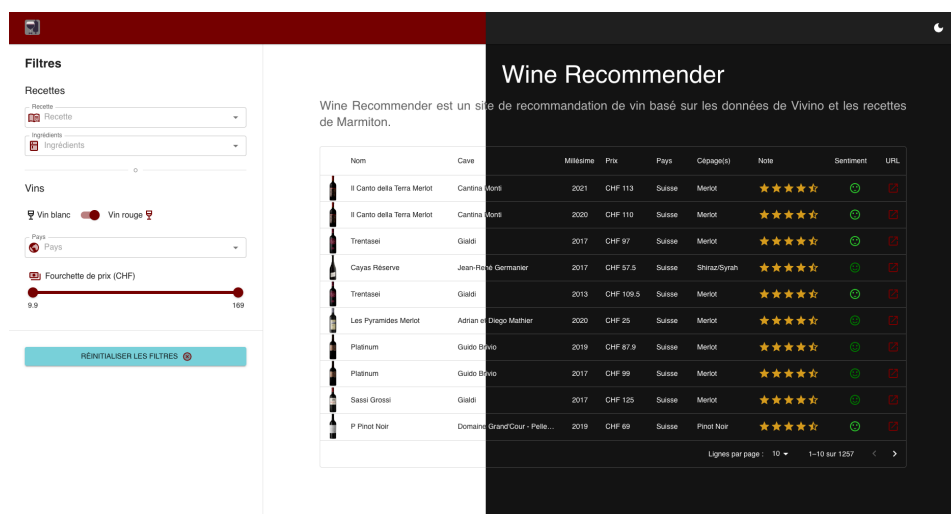
Pour le frontend, on build simplement l'application React avant de la *dockeriser* elle aussi.

Ces images sont ensuite déployées sur le Rancher de la HEIA-FR. Avec leurs propres sous-domaines :

- mongo-wine-recommender.kube.isc.heia-fr.ch (MongoDB via l'interface MongoExpress)
- api-wine-recommender.kube.isc.heia-fr.ch (Node)
- webapp-wine-recommender.kube.isc.heia-fr.ch (React)

Fonctionnalités

L'interface de base est la suivante :



Elle dispose d'une version *light* et d'une version *dark*. On peut y trouver les éléments suivants :

- Sur la droite :
 - le nom et la description de l'application
 - un tableau dynamique qui affiche les vins de la base donnée
- Sur la gauche, une zone de filtres avec les champs suivants :
 - Sélection de la recette
 - Choix des ingrédients afin de filtrer la recette
 - Vin blanc ou Vin rouge
 - Pays d'origine du vin
 - Fourchette de prix pour le vin
 - Un bouton pour réinitialiser les filtres

Filtres

Lorsque l'on modifie un des filtres pour le vin, le tableau va recharger les données en fonction de ces derniers.

Filtres

Recettes

Recette

Recette

Ingredients

Ingredients

Vins

Vin blanc

Vin rouge

Pays

FranceSuisse

Fourchette de prix (CHF)

9.9

169

REINITIALISER LES FILTRES

Wine Recommender

Wine Recommender est un site de recommandation de vin basé sur les données de Vivino et les recettes de Marmiton.

Nom	Cave	Milléisme	Prix	Pays	Cépage(s)	Note	Sentiment	URL
Cuvée Olivia Pinot Gris	Caloz	2022	CHF 25	Suisse	Pinot Gris	★★★★★	😊	🔗
Ambassadeur Fumé Gros R...	Adrian et Diego Mathier	2021	CHF 28	Suisse	Sylvaner	★★★★★	😊	🔗
Melodie	St Jodern	2022	CHF 26	Suisse	Savagnin	★★★★★	😊	🔗
Lux Vina Altinus Petite Anvine	Domaines Chevaliers	2021	CHF 38.5	Suisse	Petite Anvine	★★★★★	😊	🔗
Sélection Sous Bois Chardo...	Château d'Auvemier	2021	CHF 26.5	Suisse	Chardonnay	★★★★★	😊	🔗
Palen Réserve	Histoire d'Enfer	2020	CHF 45	Suisse	Savagnin	★★★★★	😬	🔗
Invresse	Valentina Andrei	2020	CHF 36	Suisse	Petite Anvine	★★★★★	😊	🔗
Sauvignon Blanc	Completier Kallerei	2020	CHF 71	Suisse	Sauvignon Bl...	★★★★★	😊	🔗
Sauvignon Blanc	Aagne	2022	CHF 21.8	Suisse	Sauvignon Bl...	★★★★★	😊	🔗
Porte de Novembre Johanni...	Maison Gillard	2022	CHF 19.9	Suisse	Johannisberg	★★★★☆	😬	🔗

Lignes par page : 10 1-10 sur 224

Pour la partie recette, lorsque l'on ajoute un ingrédient, une requête en arrière-plan va aller chercher, dans la base donnée, toutes les recettes qui contiennent ce dernier et retourner la liste de leurs ingrédients respectifs. Ainsi, on se retrouve tout le temps avec une liste d'ingrédients qui permet forcément d'avoir une recette.

Filtres

Recettes

Recette

Recette

Ingredients

sel 🍷 bouef haché 🍷 concoures de tomates 🍷 parmesan 🍷 sucre 🍷 huile olive 🍷 tomates 🍷 vin rouge 🍷 oignons 🍷 char à saucisse 🍷 cuise de bouillon de bouef 🍷 feuilles de laurier 🍷 mortadelle 🍷 carottes 🍷 jambon cru 🍷 poivre 🍷 spaghetti 🍷 ail 🍷 basilic 🍷 pancetta 🍷 céleri 🍷

Vins

Vin blanc

Vin rouge

Pays

Pays

Fourchette de prix (CHF)

9.9

169

REINITIALISER LES FILTRES

Wine Recommender

Wine Recommender est un site de recommandation de vin basé sur les données de Vivino et les recettes de Marmiton.

Nom	Cave	Milléisme	Prix	Pays	Cépage(s)	Note	Sentiment	URL
Il Canto della Terra Merlot	Carntina Monti	2021	CHF 113	Suisse	Merlot	★★★★★	😊	🔗
Il Canto della Terra Merlot	Carntina Monti	2020	CHF 110	Suisse	Merlot	★★★★★	😊	🔗
Trentasei	Gialdi	2017	CHF 97	Suisse	Merlot	★★★★★	😊	🔗
Cayaz Réserve	Jean-René Germanier	2017	CHF 57.5	Suisse	Shiraz/Syrah	★★★★★	😊	🔗
Trentasei	Gialdi	2013	CHF 109.5	Suisse	Merlot	★★★★★	😊	🔗
Les Pyramides Merlot	Adrian et Diego Mathier	2020	CHF 25	Suisse	Merlot	★★★★★	😊	🔗
Platinum	Guido Brivio	2019	CHF 87.9	Suisse	Merlot	★★★★★	😊	🔗
Platinum	Guido Brivio	2017	CHF 99	Suisse	Merlot	★★★★★	😊	🔗
Sassi Grossi	Gialdi	2017	CHF 125	Suisse	Merlot	★★★★★	😊	🔗
P Pinot Noir	Domaine Grand'Cour - Pelle...	2019	CHF 69	Suisse	Pinot Noir	★★★★★	😊	🔗

Lignes par page : 10 1-10 sur 1257

On peut donc en sélectionner jusqu'à ce qu'il n'en propose plus et cela nous donnera une seule recette à sélectionner.

Filtres

Recettes

Recette

Spaghetti bolognaise (proche de la vraie)

Ingredients

parmesan 🍷 sucre 🍷 huile olive 🍷 tomates 🍷 vin rouge 🍷 oignons 🍷 char à saucisse 🍷 cuise de bouillon de bouef 🍷 feuilles de laurier 🍷 mortadelle 🍷 carottes 🍷 jambon cru 🍷 poivre 🍷 spaghetti 🍷 ail 🍷 basilic 🍷 pancetta 🍷 céleri 🍷

Vins

Vin blanc

Vin rouge

Pays

Pays

Fourchette de prix (CHF)

9.9

169

REINITIALISER LES FILTRES

Wine Recommender

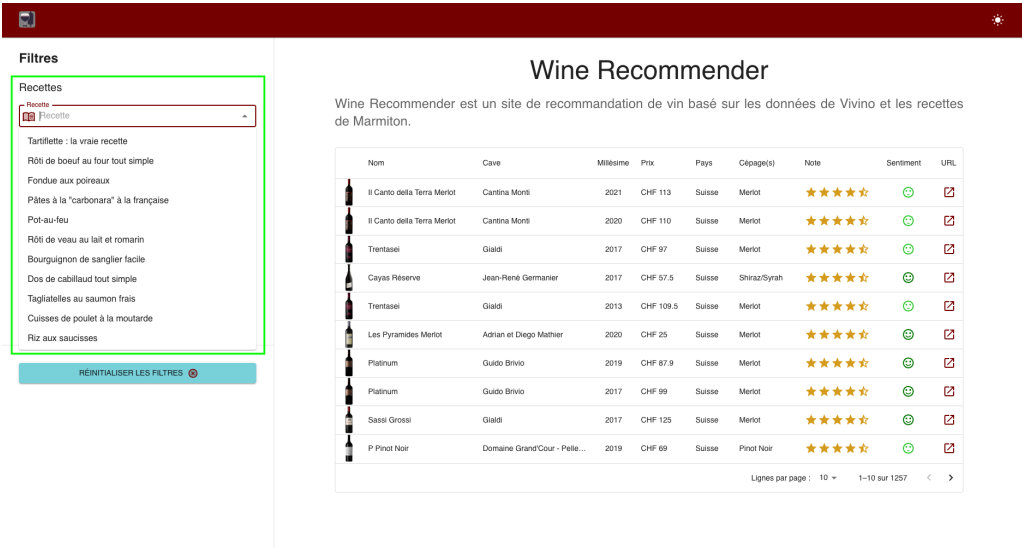
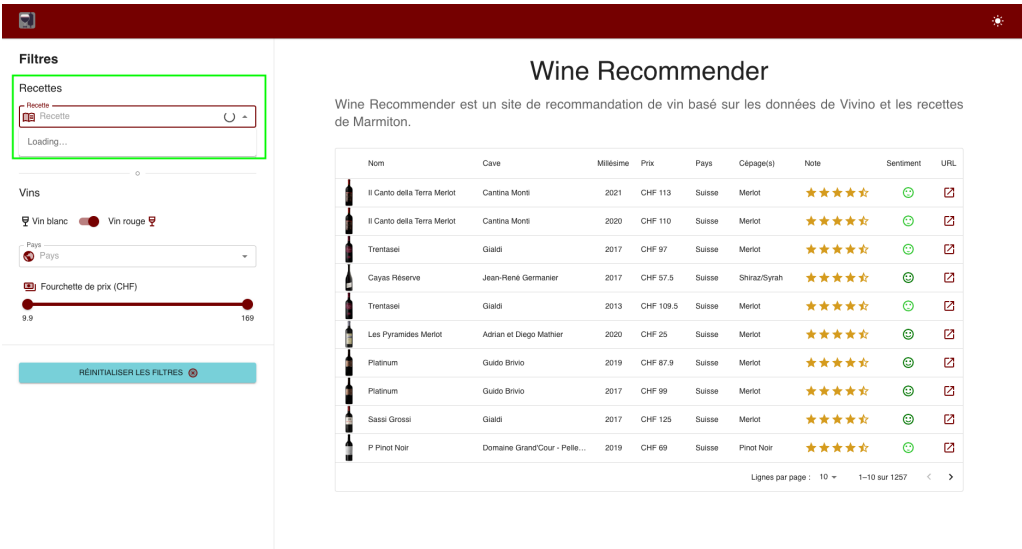
Wine Recommender est un site de recommandation de vin basé sur les données de Vivino et les recettes de Marmiton.

Nom	Cave	Milléisme	Prix	Pays	Cépage(s)	Note	Sentiment	URL
Il Canto della Terra Merlot	Carntina Monti	2021	CHF 113	Suisse	Merlot	★★★★★	😊	🔗
Il Canto della Terra Merlot	Carntina Monti	2020	CHF 110	Suisse	Merlot	★★★★★	😊	🔗
Trentasei	Gialdi	2017	CHF 97	Suisse	Merlot	★★★★★	😊	🔗
Cayaz Réserve	Jean-René Germanier	2017	CHF 57.5	Suisse	Shiraz/Syrah	★★★★★	😊	🔗
Trentasei	Gialdi	2013	CHF 109.5	Suisse	Merlot	★★★★★	😊	🔗
Les Pyramides Merlot	Adrian et Diego Mathier	2020	CHF 25	Suisse	Merlot	★★★★★	😊	🔗
Platinum	Guido Brivio	2019	CHF 87.9	Suisse	Merlot	★★★★★	😊	🔗
Platinum	Guido Brivio	2017	CHF 99	Suisse	Merlot	★★★★★	😊	🔗
Sassi Grossi	Gialdi	2017	CHF 125	Suisse	Merlot	★★★★★	😊	🔗
P Pinot Noir	Domaine Grand'Cour - Pelle...	2019	CHF 69	Suisse	Pinot Noir	★★★★★	😊	🔗

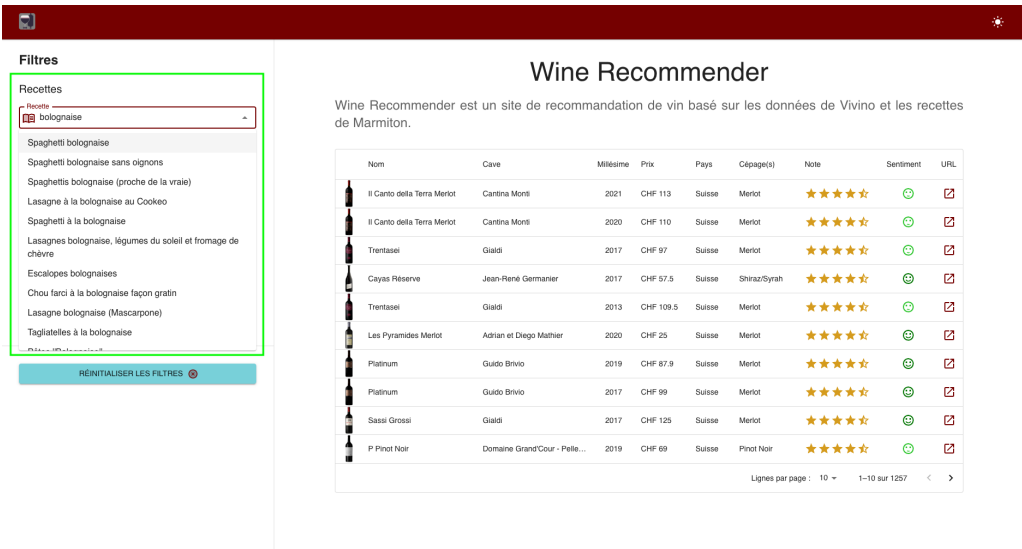
Lignes par page : 10 1-10 sur 1257

5 / 11

Concernant le champ de recherche pour les recettes, ce dernier va, dès qu'on le sélectionne, aller chercher la liste des recettes qui correspondent au filtre des ingrédients et les afficher dans le menu déroulant.



On peut ensuite commencer à écrire et la liste va s'affiner.



Finalement, lorsque l'on a trouvé la bonne recette, on peut la sélectionner. Cela va désactiver le champ ingrédients pour ne pas avoir de problème d'actualisation et rafraîchir l'interface des vins.

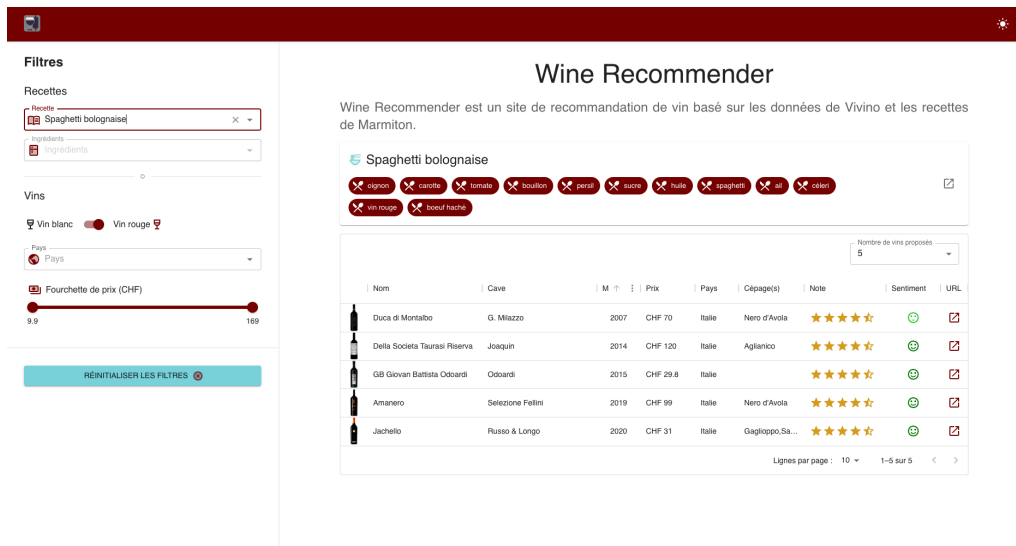


Tableau de vins

Les colonnes affichées sont les suivantes :

- l'image (en cliquant dessus, on peut le voir en plus grand)
- le nom
- la cave
- le millésime
- le prix en CHF
- le pays d'origine
- les cépages
- la note moyenne (affichée à l'aide d'étoiles et arrondie au demi)
- le pourcentage de sentiment positif (avec un smiley correspondant pour chaque tranche de 20 %)
- le lien qui mène sur le site de Vivino

Si aucune recette n'est choisie, le tableau affiche simplement tous les vins de la base de données, permettant, par exemple, de voir leur note et leur sentiment.

Affichage de la recommandation

Lorsqu'une recette est sélectionnée, cette dernière va s'afficher au-dessus du tableau des vins avec la liste de ses ingrédients ainsi qu'un lien vers la page sur Marmiton.

Le tableau, quant'à lui, va s'actualiser pour afficher les vins recommandés pour cette recette. On peut, via le menu déroulant en haut à droite du tableau, changer le nombre de vins proposés. Il est également possible d'appliquer les filtres via la barre de gauche.

Techniques, algorithmes et outils utilisés

Dans notre projet, nous utilisons trois techniques de récupération et d'analyse de données différentes :

- **Le scrapping**: pour récupérer les données que l'on a besoin sur les deux sites internet
- **Système de recommandation basé sur du "word embedding"**: pour pouvoir suggérer des vins en fonction d'une recette
- **L'analyse de sentiments**: pour analyser la "qualité" du vin en fonction des commentaires d'un vin

Scrapping

Nous utilisons trois librairies pour scraper les données : **scrapy**, **selenium** et **BeautifulSoup**. Scrapy est une librairie en Python qui permet de récupérer automatiquement des données à partir de site Web, tout comme BeautifulSoup. Nous avons utilisé BeautifulSoup lorsque Scrapy ne permettait pas de faire exactement ce que nous voulions. Ces deux librairies ont été utilisés pour récupérer uniquement les données des recettes du site Web Marmiton.

Selenium est aussi une librairie en Python qui permet d'automatiser l'interaction avec les navigateurs Web. Elle permet de simuler des actions humaines, ce qui nous permet, par exemple, de naviguer sur la page. Nous utilisons cette librairie plutôt que les deux autres citées précédemment pour récupérer les données du site Vivino, car nous avons besoin de descendre en bas de la page pour charger les données dynamiquement.

Système de recommandation basé sur du "word embedding"

Cette technique de traitement du langage naturel permet de représenter les mots sous forme de vecteurs numériques dans un espace multidimensionnel. Cette représentation permet ensuite de calculer la similarité entre les mots en fonction de leur utilisation.

Dans le cadre de ce projet, nous utilisons cette technique pour recommander des vins en fonction d'une recette. Pour ce faire, nous générons au préalable des embeddings pour les recettes et les vins en utilisant la librairie [Gensim](#) et le modèle [GoogleNews-vectors-negative300](#). GoogleNews-vectors-negative300 est un ensemble de vecteurs de mots qui ont été générés à partir des données de Google News en utilisant le modèle Word2Vec. Ces vecteurs représentent les mots sous forme de nombres réels dans un espace vectoriel de 300 dimensions. Nous utilisons ce modèle, car nous n'avons pas trouvé de modèle spécifique aux recettes ni le temps d'en entraîner un pour cette tâche. Gensim est une bibliothèque Python pour le traitement de texte et la modélisation de sujets à grande échelle, qui permet de créer des modèles de traitement de texte à partir de données brutes et de vecteurs de mots pré-entraînés, avec une grande flexibilité pour la personnalisation des paramètres.

Pour créer l'embedding des recettes, nous procédons de la manière suivante :

1. Charger les données du JSON
2. Créer un histogramme qui compte le nombre d'occurrences de chaque mot
3. Pour chaque recette, prendre le titre et les ingrédients et créer l'embedding de la manière suivante :
 - Traduire les ingrédients en anglais
 - Transformer les ingrédients en minuscules
 - Pour chaque ingrédient
 - Supprimer la ponctuation
 - Supprimer les mots qui ont une occurrence inférieure à 5 fois
 - Supprimer les stopwords (de, les ,la, le, etc...)
 - Créer l'embedding
4. Moyenner l'embedding de la manière suivante :
 - Si l'embedding des ingrédients est vide et que l'embedding du titre est vide → supprimer la recette
 - Si l'embedding des ingrédients est vide → moyenner seulement les embedding du titre
 - Si l'embedding du titre est vide → moyenner seulement les embedding des ingrédients

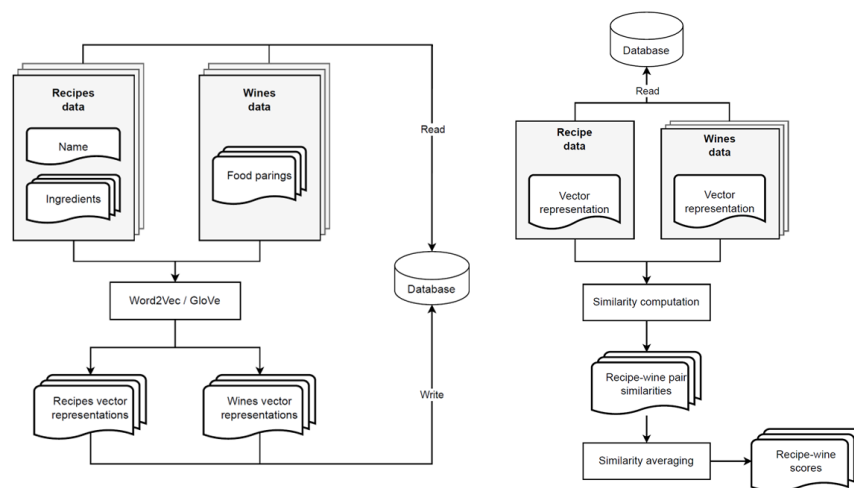
- Sinon → moyenner les deux embeddings en donnant un poids plus important au titre de la recette (dans notre cas 5)

5. Sauvegarder en JSON

Pour créer l'embedding des accords du vin, nous avons effectué les mêmes étapes en retirant la création de l'histogramme. Ces différents embeddings sont donc calculés et mis dans la base de donnée. Ce sont des données statiques qui ne vont pas changer dans le futur.

Pour ce qui est du calcul de la similarité entre un embedding d'une recette et les embeddings des vins, cette étape est faite dynamiquement en calculant la similarité cosinus entre l'embedding de la recette et celui des accords de chaque vin. Lorsque l'on a choisi une recette, une liste de quelques vins (1, 5, 10 ou 20) est affichée. Ces vins sont ceux qui ont la plus grande similarité.

L'image ci-dessous décrit le calcul des embeddings des recettes et des vins (à gauche) et le calcul des scores de similarité entre une recette spécifique et les vins de la base de données (à droite).



Analyse de sentiments

En plus de la génération d'embedding pour chaque recette et chaque vin, nous effectuons une analyse de sentiments sur les avis des vins de Vivino. Nous avons décidé de ne récupérer uniquement les trois avis qui sont mis en évidence sur la page des vins, puisqu'il s'agit des avis jugés les plus utiles par le site.

L'analyse de sentiment est un problème connu de traitement du langage naturel (NLP) et est aujourd'hui résolu principalement à l'aide de technologies de deep learning. La plupart du temps, les sentiments à prédire sont répartis en trois catégories : positif, neutre ou négatif.

Dans le cadre de ce projet, nous utilisons la librairie [pysentimiento](#) basée sur les modèles de deep learning Transformer et permettant de réaliser des tâches de NLP. Parmi ces tâches, la librairie intègre de l'analyse de sentiment, qui retourne la probabilité de chacune des trois catégories de sentiments à partir de données textuelles. Nous calculons ces probabilités pour chacun des avis des vins et les moyennons afin d'obtenir un sentiment global pour chaque vin.

L'image ci-dessous montre les résultats de l'analyse de sentiments pour chaque review d'un vin spécifique et le moyennage de ceux-ci pour obtenir le sentiment général du vin.

```
[
  {
    "review": "Robe rouge sombre, nez de petits fruits avec des éflaves poivrées et un soupçon de cuir, bouche puissante où l'on retrouve le nez avec des tanins présents mais fins, finale bien marquée. Belle bouteille.",
    "NEG": 0.0013273286167532206,
    "NEU": 0.04827291890978813,
    "POS": 0.9503996968269348
  },
  {
    "review": "N'étant absolument pas fan de syrah, j'essaye. Alors oui. Équilibré, aromatique mais pas écœurant. Magnifique travail. Pas de miracle: pour faire de bons produits, il faut un excellent domaine et y mettre le prix. Mérité. Santé. Bravo",
    "NEG": 0.007557576522231102,
    "NEU": 0.06459382176399231,
    "POS": 0.9278486371040344
  },
  {
    "review": "Nez très fruité. Belle robe grenat foncé. Agréable en bouche du début à la fin avec une belle linéarité ! À refaire !",
    "NEG": 0.0015335698844864964,
    "NEU": 0.006596689578145742,
    "POS": 0.9918696880340576
  }
]
```

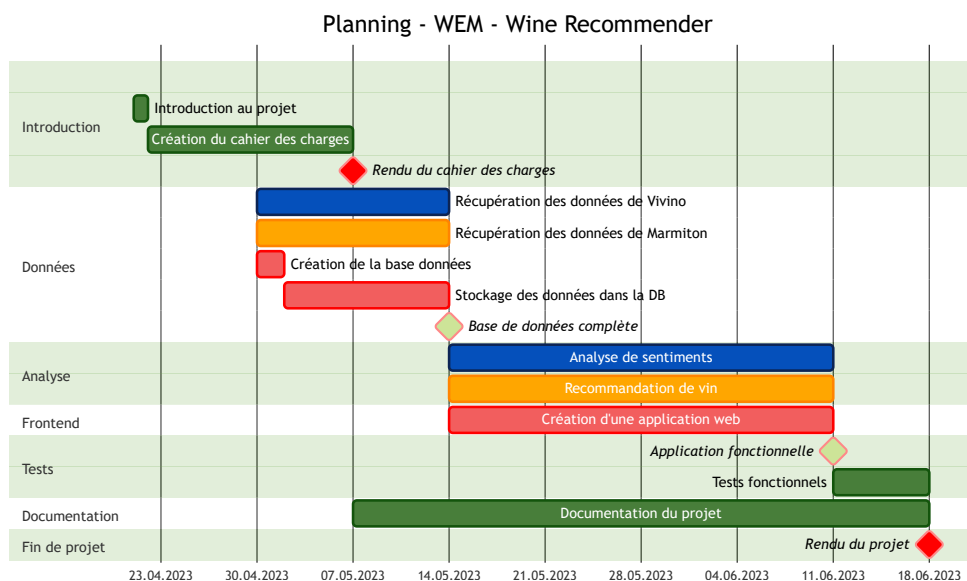
→

```
{
  "NEG": 0.0034728250078236065,
  "NEU": 0.039821143417308726,
  "POS": 0.9567060073216757
}
```

Comme mentionné précédemment, nous nous basons uniquement sur la probabilité que le sentiment soit positif pour afficher le smiley correspondant sur l'interface.

Planification, organisation et suivi, répartition du travail

La planification du projet est décrite par le diagramme de Gantt suivant :



Les couleurs indiquent la ou les personnes affectées à la tâche en question, à savoir :

- **Vert**: tout le monde
- **Bleu**: Benjamin
- **Orange**: Laurent
- **Rouge**: Andrea

Nous avons respecté ce diagramme, et toutes ces tâches ont été réalisées dans les temps. Une séance a été organisée aux dates correspondant aux jalons du projet afin de passer en revue de ce qui a été fait ainsi organiser les prochaines à effectuer.

Conclusion

Dans le cadre de ce projet, nous avons créé une application de recommandation de vins à partir de recette en appliquant les notions vues en cours pour récupérer des données automatiquement à partir de site web. De plus, nous avons également analysé les données récupérées en intégrant de l'analyse de sentiment et des calculs de similarité entre les informations des recettes et les accords mets-vins des vins. Finalement, nous avons créé une interface web fonctionnelle couplée à une API Rest et une base de données, permettant à l'utilisateur de trouver les meilleurs vins s'accordant à la recette choisie. Le tout étant déployé automatiquement à l'aide de CI/CD et disponible via des URLs.

Globalement, nous sommes très satisfaits du travail que nous avons accompli. Nous avons réussi à développer une application fonctionnelle à partir d'une problématique qui n'a beaucoup été traitée. En effet, nous n'avons trouvé qu'un seul outil semblable au nôtre et sommes même allé plus loin, en proposant plusieurs vins spécifiques pour la recette donnée par l'utilisateur. De plus, l'application des différentes notions en lien avec le Web Mining, l'analyse de données et le développement frontend nous a permis d'approfondir et/ou d'acquérir de nouvelles compétences.

Bien que nous soyons satisfaits du fonctionnement finale de notre application, il reste tout de même certains points que nous pourrions améliorer.

Travaux futurs

La qualité des résultats de similarité entre les recettes et les accords des vins est difficile à juger leur qualité mais ils sont parfois étranges. Nous avons pu apercevoir des cas où une recette de poisson s'accordait mieux avec un vin s'accordant avec de la viande plutôt qu'un vin s'accordant avec du poisson. Ce comportement peut être expliqué par le fait que des ingrédients secondaires prennent trop de poids par rapport aux principaux. En effet, les accords des vins consistent en des plats ou des ingrédients principaux (p.ex boeuf, agneau, saumon, végétarien, ...) et il serait donc préférable de donner plus de poids à ces termes lorsqu'ils apparaissent dans la recette. Nous avons déjà réglé partiellement ce problème en donnant plus de poids aux titres de la recette, qui contient le plus souvent les informations les plus importantes.

Le modèle Word2Vec utilisée et un modèle pré-entraîné sur des données de Google News et contient donc des vecteurs d'une très grande variété de mots. L'avantage est qu'il est capable de générer un vecteur pour un quantité astronomique de mots, et que nos ingrédients et accords peuvent donc quasiment tous être transformés. Cependant, il serait certainement plus bénéfique d'entraîner un modèle Word2Vec exclusivement sur des données liées à la gastronomie et à la nourriture, afin que les similarités entre les ingrédients soient plus précises.

Pour améliorer le traitement des données, il faudrait que les mots similaires (oignon et oignons) soient considérés comme le même mot. Une solution serait par exemple d'utiliser stemming, qui permettrait certainement d'améliorer les résultats des similarités calculées entre les embedding. De plus, ceci permettrait d'éviter des doublons dans la liste des ingrédients à choisir sur l'interface Web.

Pour l'architecture, nous pourrions ajouter des sécurités sur l'API afin de ne pas avoir accès directement à la DB.

Une dernière amélioration (peut-être utopique), serait d'effectuer les différentes requêtes et inférences dynamiquement sur les sites afin de toujours avoir des données à jour.