# Note méthodologique : preuve de concept

## Dataset retenu

2 Data Set :

1 - **Projet 3** :  Santé Publique France (pour l'évaluation)
2 - **TheFoodProcessors**: https://huggingface.co/Thefoodprocessor

1 - **Projet 3 :** Le jeu de données d'OpenFoodFacts contient l'ensemble des produits répertoriés, soit 320 772 individus. Nous observons 182 colonnes, et nous allons particulièrement utiliser et analyser les colonnes 'product_name', 'ingredient_text' et 'allergens'.
Mon travail sur ce projet a consisté à extraire les individus d'origine française, créant ainsi un sous-ensemble nommé *data_france*. Sur cet ensemble, j'ai appliqué un premier traitement des erreurs de saisie afin d'observer s'il pouvait exister un lien entre la présence d'allergènes et le Nutri-Score.
**Avantages :** Grande quantité de données, avec de nombreux produits et autant de listes d'ingrédients. Présence de produits internationaux, apportant une grande diversité dans les recettes.
**Inconvénients :** La saisie manuelle des ingrédients et allergènes crée de nombreuses erreurs. La correction est complexe, introduisant des biais, des choix arbitraires et une perte d'information. De plus, le caractère international du jeu de données pose des défis liés à la diversité des langues et alphabets utilisés. Les réglementations d'affichage des ingrédients varient également d'un pays à l'autre.

Dans son état actuel, même après traitement, ce jeu de données ne permet pas un entraînement efficace pour le modèle.

2 - **TheFoodProcessors/allergy_type :** Ce jeu de données, disponible sur la plateforme open source Hugging Face, contient 74 465 individus avec trois colonnes, dont une comprenant la recette et une autre les allergènes présents. Nous allons nous concentrer sur 9 allergènes : milk, egg, nut, wheat, soy, gluten, fish, peanut, seafood. Ces allergènes révéleront différents problèmes techniques, comme les variations de représentation, la proximité des termes et les contextes de leur apparition.
**Avantages :** Jeu de données particulièrement bien adapté à notre projet, avec un grand volume de données. Les recettes sont rédigées de manière

contextualisée.
**Inconvénients :** Le jeu de données est uniquement en anglais.

Nous allons utiliser ce jeu de données pour entraîner notre modèle, mais il faudra réfléchir à la possibilité de traduire les données.

## Les concepts de l'algorithme récent

Dans le jeu de données d'Open Food Facts, de nombreux allergènes ont été saisis manuellement, entraînant un grand nombre d'erreurs, tant orthographiques que de compréhension. Cela pose des risques significatifs pour les utilisateurs et les consommateurs qui pourraient prendre les informations de l'application comme exactes. Pour remédier à ce problème, nous proposons de mettre en place un algorithme de détection automatique des allergènes en analysant une liste d'ingrédients ou une recette.

Pour la preuve de concept, nous avons choisi d'utiliser un modèle **Bi-LSTM (Bidirectional Long Short-Term Memory)** avec TensorFlow/Keras. Cet algorithme nous permettra d'identifier les mots-clés qui se rapportent à un allergène spécifique.

**Fonctionnement du Bi-LSTM**

Le Bi-LSTM est une extension des réseaux de neurones récurrents (RNN) traditionnels, conçue pour traiter les données séquentielles telles que le texte. Voici comment il fonctionne :

- **Traitement bidirectionnel** : Contrairement aux RNN classiques qui traitent les données dans une seule direction (du début à la fin), le Bi-LSTM traite les séquences dans les deux sens. Cela signifie qu'il analyse le contexte précédent et suivant de chaque mot dans une phrase, améliorant ainsi la compréhension contextuelle.
- **Mécanisme de mémoire à long terme** : Les LSTM (Long Short-Term Memory) sont dotés de cellules mémoire capables de conserver des informations sur de longues séquences. Elles utilisent des portes (input, output et forget gates) pour contrôler le flux d'informations, ce qui permet de résoudre le problème du gradient qui disparaît souvent rencontré dans les RNN standard.
- **Capturer les dépendances à long terme** : Grâce à sa structure, le Bi-LSTM peut capturer les relations complexes entre les mots, même s'ils sont éloignés dans la séquence. Cela est particulièrement utile pour identifier les allergènes qui peuvent être mentionnés sous différentes formes ou dans des contextes variés.

**Pourquoi utiliser le Bi-LSTM pour la détection des allergènes ?**

- **Précision accrue** : En tenant compte du contexte complet des ingrédients, le Bi-LSTM peut distinguer entre des termes similaires et identifier précisément les allergènes.
- **Gestion des variations linguistiques** : Les erreurs orthographiques et les variations de langage courantes dans les saisies manuelles peuvent être mieux gérées grâce à la capacité du Bi-LSTM à apprendre des représentations contextuelles.
- **Efficacité opérationnelle** : En automatisant la détection, nous réduisons la dépendance à la saisie manuelle, minimisant ainsi les erreurs humaines et accélérant le processus de mise à jour des données.

**Perspectives futures avec le modèle ELECTRA**

Une fois le concept validé avec le Bi-LSTM, nous pourrions envisager d'utiliser le modèle **ELECTRA** pour la phase de mise en application. ELECTRA est un modèle de langage pré-entraîné qui offre des performances supérieures pour les tâches de compréhension du langage naturel. Bien qu'il soit plus coûteux en termes de ressources computationnelles, il pourrait améliorer encore la précision et la robustesse de la détection des allergènes.

**Bénéfices attendus**

- **Amélioration de la qualité des données** : Des données plus précises permettent non seulement de protéger les consommateurs, mais aussi de faciliter d'autres travaux futurs basés sur ces données fiables.
- **Expérience utilisateur optimisée** : En éliminant le besoin de saisie manuelle, l'application devient plus conviviale et accessible.
- **Sécurité accrue** : En fournissant des informations exactes sur les allergènes, nous contribuons à réduire les risques pour les consommateurs sensibles ou allergiques.

# La modélisation

Nous avons utilisé un jeu de données accessible via Hugging Face, comprenant **74 500 recettes** associées à leurs allergènes respectifs. Ce jeu de données a servi de base d'entraînement pour notre modèle. Dans un premier temps, nous avons extrait **neuf allergènes spécifiques** : *milk, egg, nut, wheat, soy, gluten, fish, peanut, seafood.*

**Prétraitement des Données**

Pour préparer les données textuelles des recettes, nous avons appliqué plusieurs étapes de prétraitement :

- **Nettoyage du Texte** : Suppression de la ponctuation et des nombres à l'aide du module **`re`** (expressions régulières) pour réduire le bruit dans les données.
- **Gestion des Stopwords** : Élimination des mots vides en anglais en combinant les listes de stopwords de **NLTK** et de **scikit-learn**. Nous avons pris soin de ne pas supprimer les mots-clés des allergènes en les excluant de la liste des stopwords.
- **Lemmatisation** : Utilisation du **`WordNetLemmatizer`** de **NLTK** pour réduire les mots à leur forme de base, normalisant ainsi le texte pour une meilleure analyse.

**Bibliothèques Clés Utilisées**

- **TensorFlow et Keras** : Utilisés pour construire et entraîner le modèle de réseau de neurones LSTM, permettant de capturer les dépendances séquentielles dans les données textuelles.
- **NLTK (Natural Language Toolkit)** : Employé pour le traitement du langage naturel, notamment pour la lemmatisation et la gestion des stopwords.
- **Scikit-learn** : Fournit des stopwords supplémentaires et le **`MultiLabelBinarizer`**, facilitant la gestion de la classification multi-étiquettes.
- **Joblib** : Permet le chargement efficace d'objets Python tels que le tokenizer et le `MultiLabelBinarizer`.
- **re** : Module d'expressions régulières utilisé pour le nettoyage du texte.

**Approches de Modélisation**

Pour déterminer la meilleure approche de modélisation, nous avons exploré différentes techniques d'apprentissage automatique :

- **Régression Logistique**
- **TensorFlow et Keras**
- **Naïve Bayes**
- **Réseaux de Neurones LSTM**

En raison de limitations matérielles, nous n'avons pas pu mettre en œuvre des modèles plus complexes tels que **BERT**, **ELECTRA** ou **GPT-3**, bien que leur potentiel ait été considéré théoriquement.

**Évaluation des Modèles**

Pour évaluer les performances de chaque modèle, nous avons utilisé plusieurs métriques :

- **Accuracy de Validation**
- **Précision (Precision)**
- **Rappel (Recall)**
- **F1-score**
- **Support** pour chaque allergène

De plus, nous avons soumis une série de cinq recettes au modèle sauvegardé pour observer les résultats en conditions réelles. Cela nous a permis d'analyser le taux de prédiction brute et de déterminer manuellement le seuil nécessaire pour l'affichage d'un allergène.

**Résultats Préliminaires**

Les premiers tests indiquent que l'utilisation de **TensorFlow et Keras associée à un réseau Bi-LSTM** est la plus prometteuse pour notre preuve de concept. Le modèle LSTM a été entraîné sur les séquences textuelles prétraitées, en utilisant un **tokenizer** pour convertir le texte en séquences numériques et en appliquant un **padding** pour uniformiser la longueur des séquences.

Cependant, plusieurs défis ont été identifiés :

- **Déséquilibre des Données** : La répartition des allergènes n'est pas équilibrée dans le jeu de données, ce qui affecte la performance globale du modèle, notamment pour les allergènes rares.
- **Prédictions Incorrectes** : Le modèle a des difficultés à prédire correctement certains allergènes évidents. Par exemple, certains allergènes présents dans les recettes ne sont pas détectés.
- **Faible Taux de Prédiction Brute** : Le taux actuel est trop bas pour être fiable sans intervention manuelle. Un seuil de décision a été fixé (par exemple, 0,27) pour convertir les probabilités en prédictions binaires, mais ce seuil nécessite un ajustement pour améliorer la précision.

Pour pallier ces limitations, nous avons combiné les prédictions du modèle LSTM avec une approche basée sur la recherche de mots-clés dans les recettes. En vérifiant la présence des allergènes dans le texte prétraité, nous

avons pu améliorer la détection globale en fusionnant les résultats des deux méthodes.

## Une synthèse des résultats

Dans le point précédent, nous avons constaté que l'entraînement des différents modèles et l'optimisation du LSTM révélaient certains problèmes. Pour y remédier, plusieurs approches ont été mises en œuvre.

**Rééquilibrage des données :**

Dans un premier temps, j'ai rééquilibré les données en créant un nouveau jeu plus précis, avec une représentation minimale et maximale des allergènes ciblés. Le premier entraînement a été effectué avec une répartition très équilibrée. Cependant, l'efficacité du modèle à identifier l'allergène "milk" est devenue très mauvaise, alors qu'elle était excellente précédemment. Ce constat a mis en évidence que "milk" est très présent dans des recettes aux ingrédients beaucoup plus diversifiés. Sa détection est donc plus complexe et nécessite une représentation plus importante dans le jeu de données, tout comme pour l'allergène "fish". J'ai donc ajouté une contrainte pour contrôler la représentation maximale de ces allergènes dans les recettes.

**Mise en place d'une deuxième étape d'analyse :**

Ensuite, j'ai décidé d'introduire une deuxième étape dans la détection des allergènes. Le principe est le suivant : **si un allergène est clairement mentionné dans la recette, alors il est considéré comme présent**. Cette étape permet de détecter certains allergènes explicitement indiqués dans la recette, mais qui pourraient ne pas être identifiés par le modèle en raison du contexte.

Ces deux manipulations ont permis d'améliorer le taux de prédictions, mais le modèle restait fragile.

**Double entraînement du modèle :**

Pour renforcer la robustesse du modèle, j'ai exploré une autre piste d'amélioration consistant à entraîner le modèle deux fois :

1. **Premier entraînement** sur toutes les recettes contenant l'allergène "milk".
2. **Deuxième entraînement** sur les huit autres allergènes.

Ainsi, nous disposons de **trois étapes de vérification** :

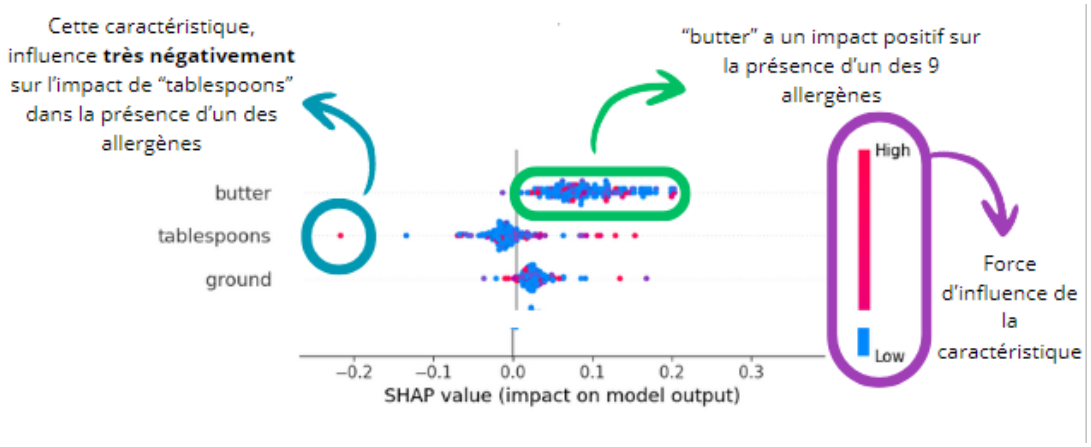
- **Présence de "milk" :** Le modèle vérifie si "milk" est présent dans la recette.
- **Présence d'un autre allergène :** Le modèle détecte les autres allergènes potentiels.
- **Mention explicite d'un allergène :** Si un allergène est clairement inscrit dans la recette, il est automatiquement considéré comme présent.

En adoptant des paramètres d'entraînement de base, nous avons obtenu un temps d'entraînement plus court pour les deux modèles et un taux de prédiction brut plus élevé. **Cette approche s'avère pour le moment la plus intéressante**.

## L'analyse de la feature importance globale et locale du nouveau modèle

Nous avons mis en place SHAP (SHapley Additive exPlanations) pour comprendre comment notre modèle prédit la présence d'allergènes dans les recettes. SHAP est un outil puissant qui permet d'interpréter les caractéristiques ayant un impact significatif sur les prédictions du modèle, en fournissant des valeurs explicatives qui révèlent l'influence de chaque caractéristique sur le résultat final. En utilisant SHAP, nous pouvons générer différents types de graphiques qui visualisent les biais potentiels et mettent en évidence des pistes d'amélioration pour le modèle.

Dans notre situation, nous avons paramétré SHAP avec K-Means pour observer plusieurs clusters de caractéristiques impactantes sur la décision de présence d'allergènes ou non. Pour ce faire, nous avons analysé un échantillon de 175 recettes, représentant 0,42% des données utilisées pour l'entraînement du modèle. Cette approche nous a permis de segmenter les données et de mieux comprendre comment certaines caractéristiques influencent les prédictions, en identifiant des schémas ou des tendances dans l'utilisation des ingrédients.



Cependant, malgré la richesse des observations possibles, la pertinence de ces résultats est questionnable. Les modèles de deep learning, tels que le Bi-LSTM que nous utilisons, sont très complexes et nécessitent des outils d'interprétation robustes, souvent coûteux en termes de ressources computationnelles. Le temps de calcul devient un facteur non négligeable dans notre analyse. Dans notre cas, l'analyse des 175 recettes a nécessité un temps de calcul très proche de celui nécessaire à l'entraînement du modèle, ce qui soulève des questions sur l'efficacité de l'utilisation de SHAP dans ce contexte.

Bien que nous ayons observé que l'impact du terme "butter" correspondait à nos attentes, les résultats obtenus n'apportent pas d'informations suffisamment précises pour être exploitées de manière concrète. Dans le cadre d'une démarche de preuve de concept, il semble peu cohérent de consacrer plus de temps à l'exploitation des résultats de SHAP, alors que des métriques plus simples.

Cependant, il est important de noter qu'une fois que le modèle sera mis en production et qu'il sera utilisé dans des contextes plus complexes, l'utilisation de SHAP pourrait s'avérer être un atout précieux. Les visualisations offertes par SHAP permettraient alors de finaliser l'amélioration du modèle en résolvant des problèmes plus complexes, tels que l'identification des biais de données ou l'analyse de l'impact des ingrédients moins courants. De plus, la compréhension des caractéristiques qui influencent les prédictions pourra guider des ajustements dans le modèle, permettant ainsi d'améliorer sa performance globale.

En résumé, l'intégration de SHAP dans notre processus d'analyse de la feature importance fournit une perspective intéressante sur les décisions du modèle, mais elle doit être mise en balance avec le temps de calcul et l'efficacité opérationnelle. À mesure que nous progressons dans l'amélioration du modèle et que nous abordons des questions plus complexes, SHAP pourra devenir un outil fondamental pour explorer et interpréter les résultats de manière approfondie.

## Les limites et les améliorations possibles

Actuellement, la performance brute de prédiction de notre modèle se situe autour de **0,3**, ce qui est insuffisant. Pour atteindre un niveau de satisfaction acceptable, nous visons une performance supérieure à **0,5**.

Cependant, plusieurs possibilités d'amélioration intéressantes s'offrent à nous. Nous avons constaté que diviser les entraînements des modèles pour améliorer leurs capacités sur un allergène plus complexe, en l'isolant d'un groupe,

augmentait considérablement l'efficacité de l'apprentissage. Il serait donc judicieux de répéter cette approche pour d'autres allergènes.

Nous pouvons également envisager l'utilisation de modèles plus performants, notamment **ELECTRA**, qui offre de meilleurs résultats avec moins de données d'entraînement comparé à **BERT**. De plus, ELECTRA est plus rapide à entraîner que les modèles traditionnels de la famille des transformeurs.

L'utilisation de **GPT** pourrait nous aider à enrichir notre jeu de données. Nous pourrions effectuer les traductions nécessaires et générer des recettes supplémentaires afin d'ajouter des exemples où certains allergènes sont sous-représentés.

À l'avenir, élargir l'entraînement à un plus grand nombre d'allergènes nous permettrait d'observer et de prévenir les allergies croisées (par exemple, le latex, le kiwi et la banane sont souvent liés). Il sera également nécessaire d'entraîner le modèle sur des cas particuliers pour éviter les erreurs liées aux combinaisons de mots. Par exemple, dans le cas du "lait d'amande", l'allergène lait n'est pas présent.

En mettant en œuvre ces améliorations, nous espérons augmenter la performance et l'interprétabilité de notre modèle, tout en répondant mieux aux besoins spécifiques liés à la prédiction des allergènes.