RAPPORT PROJET

TRANSPARENCE CLASSIFICATION DES PRODUITS ALIMENTAIRES

Noms des membres du groupe:

Ahmed Nadir Bounoua, Youcef Bensid, Anis Si Youcef, Julien VU.

Introduction

Dan

A: MODÈLE ELECTRE-TRI

Présentation du modèle:

Electre Tri est un algorithme d'aide à la décision multicritère qui permet d'affecter une classe à un élément, en se basant sur sa concordance globale face à des profils et des poids définis.

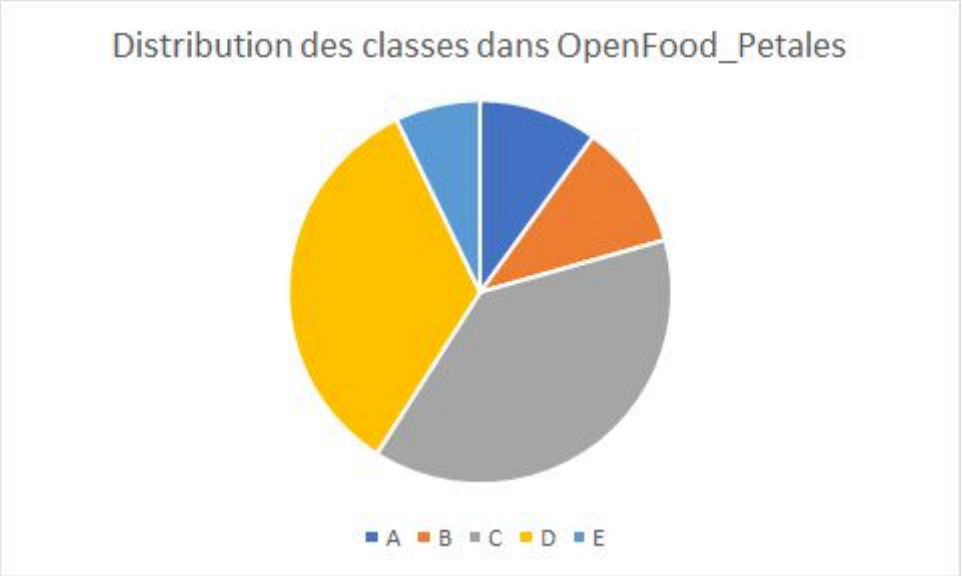
On a implémenté deux versions de cet algorithme : l'approche optimiste et l'approche pessimiste.

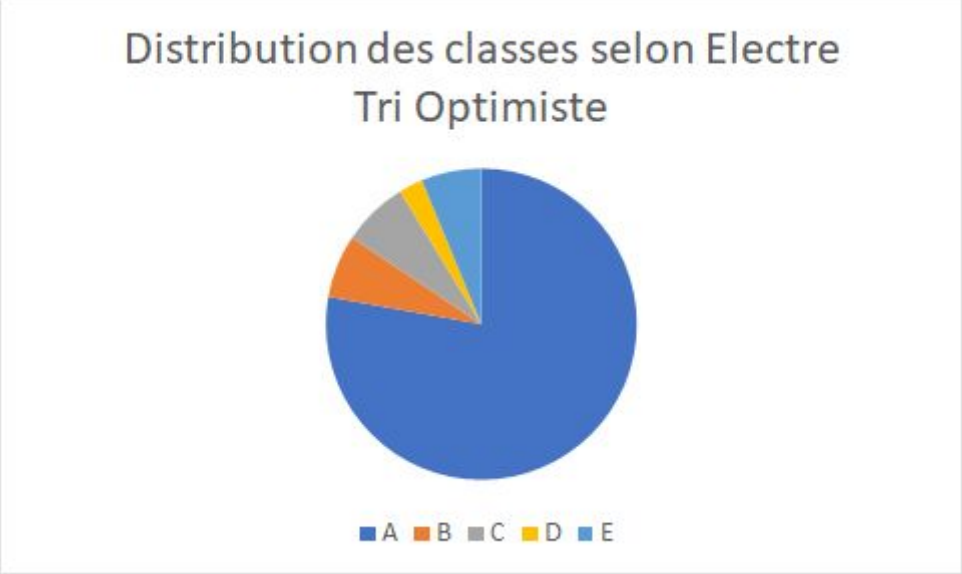
Ces algorithmes sont appliqués au NutriScore, un score nutritionnel qui donne une classe à des aliments : de la classe A pour les plus nutritifs jusqu'à la classe E pour les moins nutritifs.

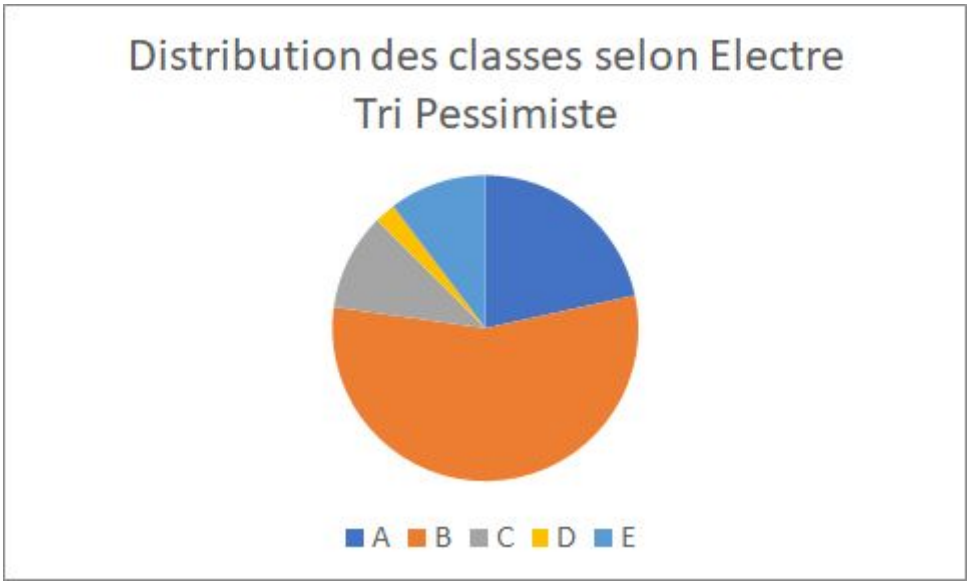
Après avoir appliqué ces algorithmes à la base de donnée "OpenFood_Petales" on a obtenu les résultats suivants.

Analyse du modèle :

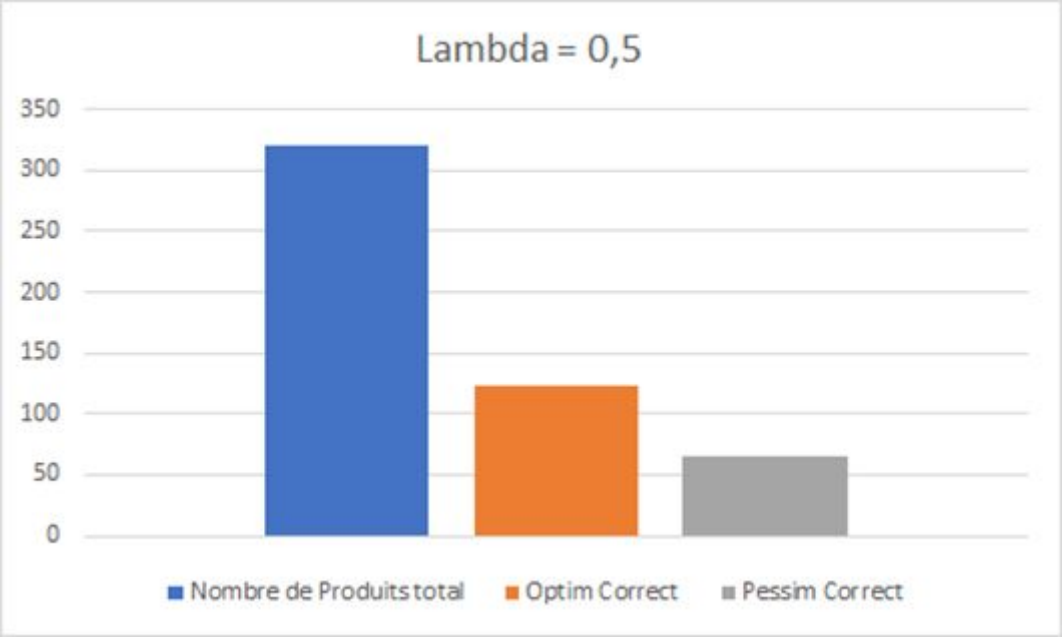
On a commencé par analyser la répartition des classes donnée par nos algorithmes, on remarque qu' Electre Tri optimiste a tendance à classer les produits dans la classe A alors que la version pessimiste de cet algorithme a tendance à classer les aliments dans la classe B, ces remarques sont faites à partir des graphiques suivants :

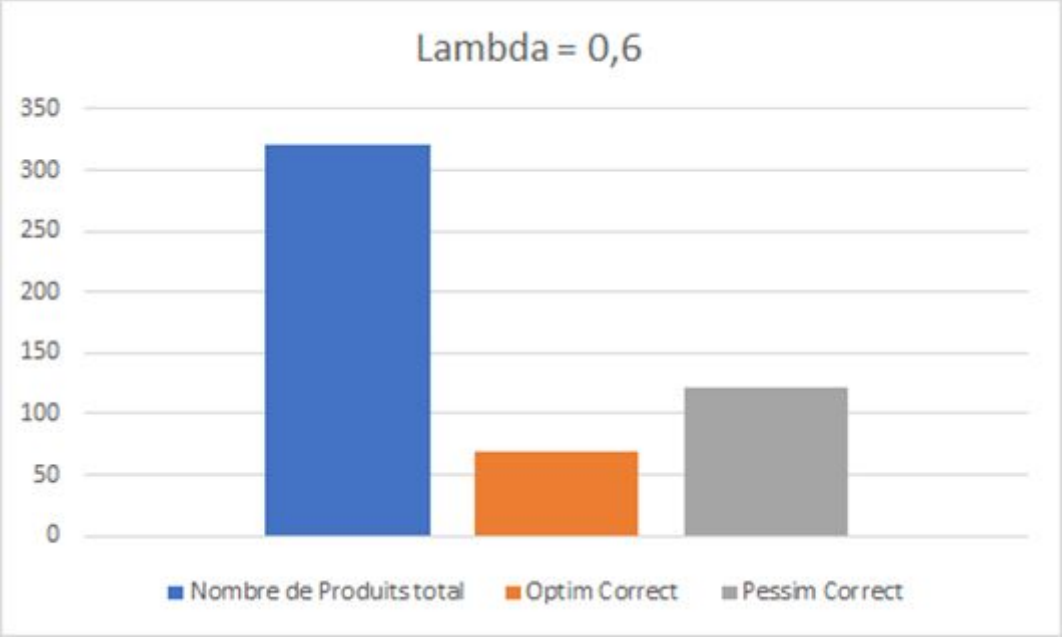

Distribution des classes dans OpenFood_Petales
A B C D E

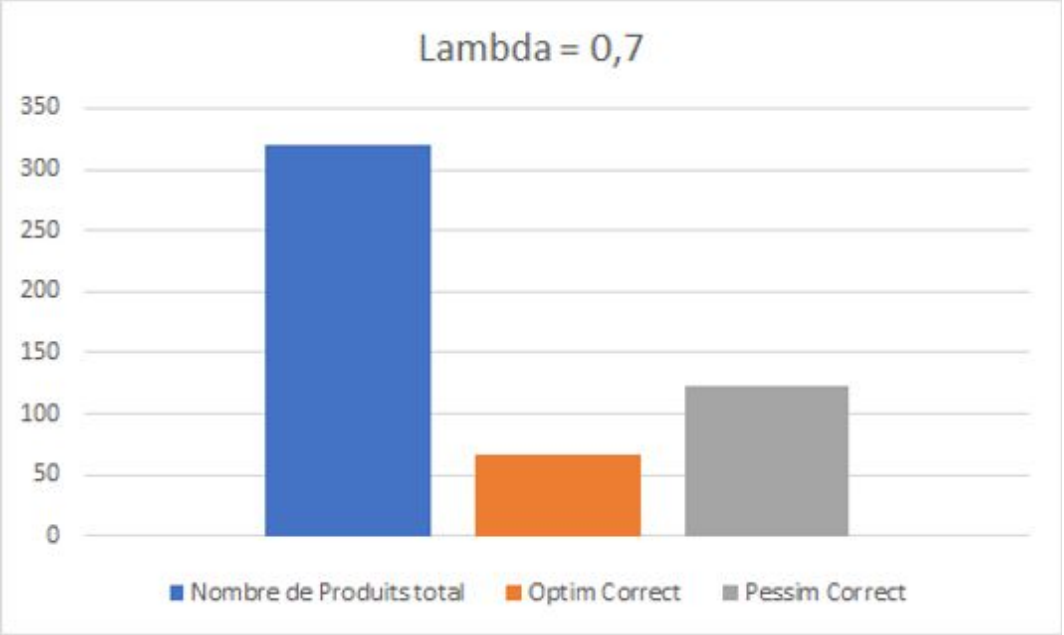

Distribution des classes selon Electre
Tri Optimiste
A B C D E


Distribution des classes selon Electre
Tri Pessimiste
A B C D E

Modification du lambda et son impact sur la classification des produits :







On remarque que modifier le lambda, qui est un paramètre de l'algorithme Electre Tri, donne des résultats différents : lorsque le lambda est égal à 0.5 c'est la version optimiste de l'algorithme qui donne une bonne classification des produits, et lorsque lambda >0.5, c'est la version pessimiste qui donne de meilleurs résultats. Cela dit, la classification dans les deux modèles reste loin de donner des résultats identiques à la classification du NutriScore. C'est pour cela qu'on a déterminé de nouveaux profils et de nouveaux poids de la manière suivante.

Nouveaux poids et profils :

Pour déterminer de nouveaux poids, on s'est basé sur le principe même du NutriScore : pourquoi a t il été créé, la réponse est la suivante : "aider les consommateurs à acheter les aliments de meilleure qualité nutritionnelle". D'un autre côté on sait que la majorité des consommateurs qui cherchent à améliorer leur habitudes alimentaires sont des personnes qui surveillent leurs lignes ou qui sont soumises à des régimes alimentaires stricts qui limitent la consommation de certains nutriments (pour la perte de poids, ou pour des raisons médicales pour ne citer que deux exemples). En se basant sur cette analyse et sur la définition même du NutriScore on arrive à la conclusion suivante : Il serait préférable de donner plus de poids aux critères à minimiser par rapport aux critères à maximiser. On obtient donc les **poids** suivants :

Énergie	Acide Gras saturé	Sucre	Fibre	Protéine	Sodium
2	2	2	1	1	2

L'idée derrière le choix des nouveaux profils est la suivante : garder les profils b1 et b6 car ils ne sont jamais atteints et donc ils représentent bien les limites. Quant aux

reste des profils on a parcouru la base de données “OpenFood_Petales”, et pour chaque classe on a pris le profil qui avait le pire NutriScore associé à une classe donnée.

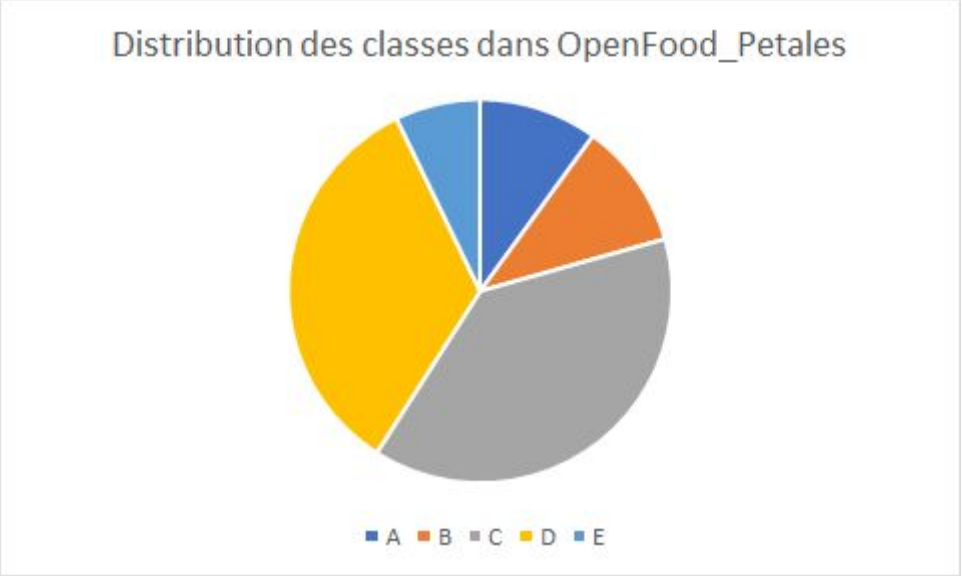
Exemple : le produit “Crosti Petales Choc” est affecté à la classe ‘A’ avec un NutriScore égal à -1, vu que le NutriScore pour la classe A va de -15 pour un très bon produit à -1 pour le plus mauvais produit (toujours de la classe A), alors le profil b5 est représenté par ce produit la.

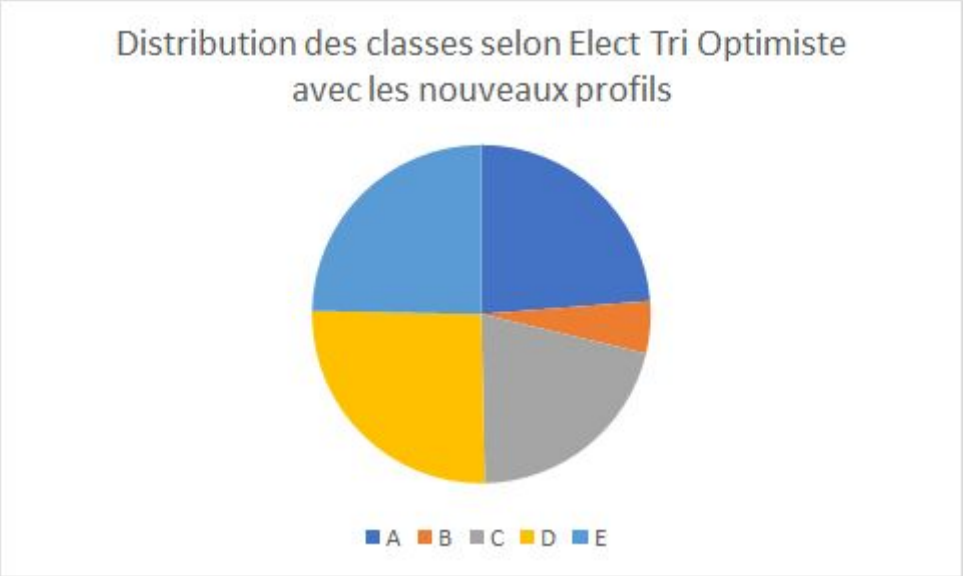
On obtient les nouveaux profils suivants :

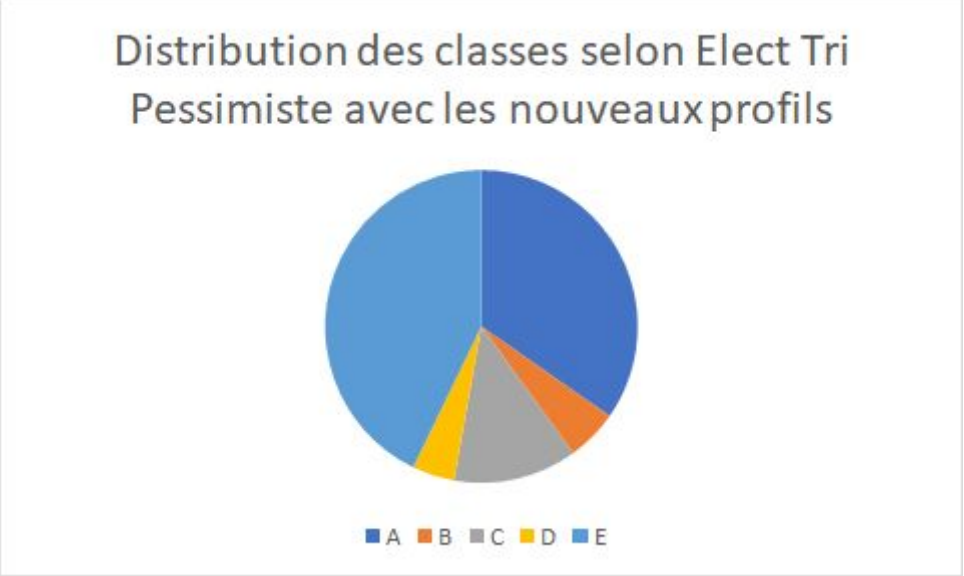
Profils	Énergie	Acide Gras saturé	Sucre	Fibre	Protéine	Sodium
b6	100	0	0	100	100	0
b5	1623	2,4	18	9	11	0,08
b4	1615	0,2	6,7	2,3	8,8	0,44
b3	1628	1,4	27,2	2,1	8,1	0,128
b2	1187	5,6	0,5	0	34	2,16
b1	10000	100	100	0	0	100

Analyse du nouveau modèle:

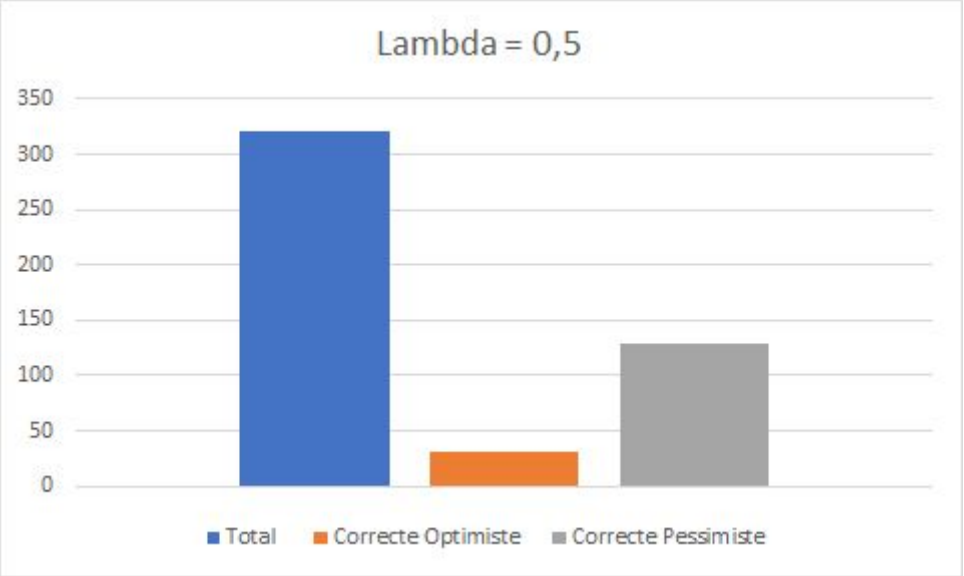
On remarque qu’avec les nouveaux profils et poids définis, les résultats sont plus parcimonieux et ne favorisent pas une classe par rapport à une autre.

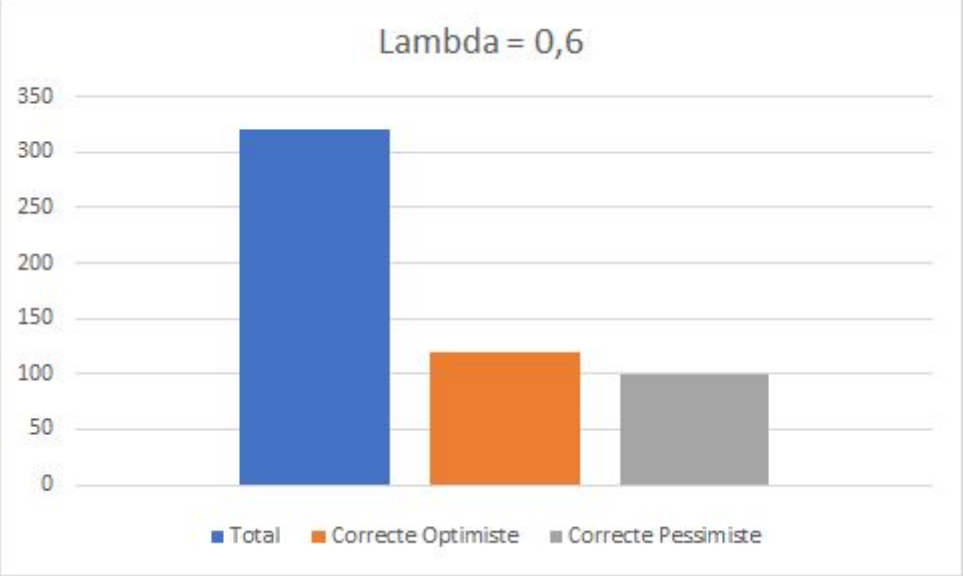


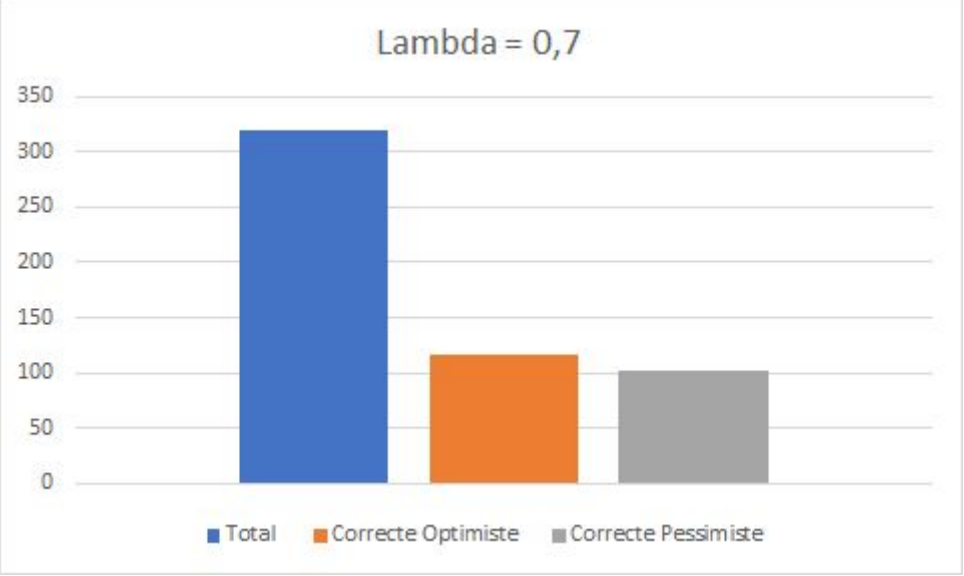




Modification du lambda et son impact sur la classification des produits selon l'algorithme Electre Tri avec les nouveaux profils et poids définis:


Lambda = 0,5
350
300
250
200
150
100
50
0
Total
Correcte Optimiste
Correcte Pessimiste


Lambda = 0,6
350
300
250
200
150
100
50
0
Total
Correcte Optimiste
Correcte Pessimiste


Lambda = 0,7
350
300
250
200
150
100
50
0
Total
Correcte Optimiste
Correcte Pessimiste

On remarque que Electre Tri avec les nouveaux profils et poids donne de meilleurs résultats que l'ancienne version de cet algorithme, en effet le nombre de classifications correctes pour, par exemple, lambda = 0.7 est égal à 117, pour la version optimiste de l'algorithme, c'est équivalent à une accuracy de 36.5%, qui est supérieure à l'accuracy d'Electre Tri avec les anciens poids pour le même lambda (0.7) et la même version de l'algorithme : 10,6%.

B: MODELE FEUX TRICOLORES

Présentation du modèle

Le modèle des feux tricolores permet de donner une étiquette à chaque produit alimentaire, l'étiquette est constituée de quatre couleurs, une couleur pour chacune des quatre composantes alimentaires suivantes : quantité de gras, graisses saturées, sucre et sel.

Contrairement aux autres modèles développés, ce modèle a une accuracy de 100%, car son fonctionnement est très simple : il fait une comparaison avec les seuils définis, et selon cette comparaison il affecte une couleur : rouge, orange ou vert.

Produits	Quantité faible	Quantité modérée	Quantité élevée
Lipides	<3	>3 et <20	>20
Acides gras saturés	<1,5	>1,5 et <5	>5
Sucres	<5	>5 et <12,5	>12,5
Sel	<0,3	>0,3 et <1,5	>1,5

C: MODÈLE NOVA

La classification NOVA consiste à classifier les produits alimentaires en regardant le degré de transformation des produits. Les produits alimentaires peuvent être répartis dans 4 groupes:

1-Groupe 1: Aliments non transformés ou transformés minimalement
2-Groupe 2: Ingrédients culinaires transformés
3-Groupe 3: Aliments transformés
4-Groupe 4: Produits alimentaires et aliments ultra-transformés

Pour classifier les produits alimentaires selon la classification NOVA, nous avions décidé de les classifier en utilisant les arbres de décision de type classification.
Les arbres de décision sont des outils d'aide à la décision qui sont très répandus dans l'apprentissage automatique. Il s'agit d'une méthode supervisée. Ils offrent la possibilité de visualiser graphiquement un ensemble de choix possibles sous la forme d'un arbre. Concrètement, chaque arbre modélise une hiérarchie de tests pour évaluer un résultat. En effet, dans un arbre de décision, nous avons des noeuds et des branches. Les décisions possibles sont situées aux niveaux des branches de l'arbre selon les décisions prises à chaque étape. C'est donc un processus itératif dans la mesure où on répète les tests jusqu'à ce qu'on trouve une classification des éléments optimale.
Ces arbres ont pour objectif de prédire à quelle classe la variable de sortie appartient. Dans notre étude, il s'agit de prédire la classe NOVA d'un ensemble de produits qui proviennent d'une des bases de données d'open food facts BD2 et BD3.
Dans le cadre de notre arbre de décision de classification, nous avions eu besoin de sélectionner certaines caractéristiques des produits qui semblaient déterminants dans notre modèle. Il s'agit de la catégorie du produit, de la sous-catégorie du produit et du nombre d'additifs qui est contenu dans le produits. Il existe 11 catégories de produits. Le nombre de sous-catégories est variable selon la catégorie du produit étudiée. Une phase de conversion des données deux premières caractéristiques est nécessaire car les arbres de décision fonctionnent uniquement avec des données quantitatives. Ainsi, un produit très bien classé aura potentiellement un nombre d'additifs très faible. La variable de sortie est donc la classe NOVA associée à chaque produit. L'autre phase importante qu'on a fait est de pouvoir séparer les données d'apprentissage et les données tests qui serviront à évaluer la précision de notre modèle. On a 70% de données d'apprentissage et 30 % de données à prédire.

Au niveau des résultats, nous avions pu fournir ci-dessous un schéma d'arbre de décision pour un nombre de noeuds égal à 8 et de profondeur égal à 8.

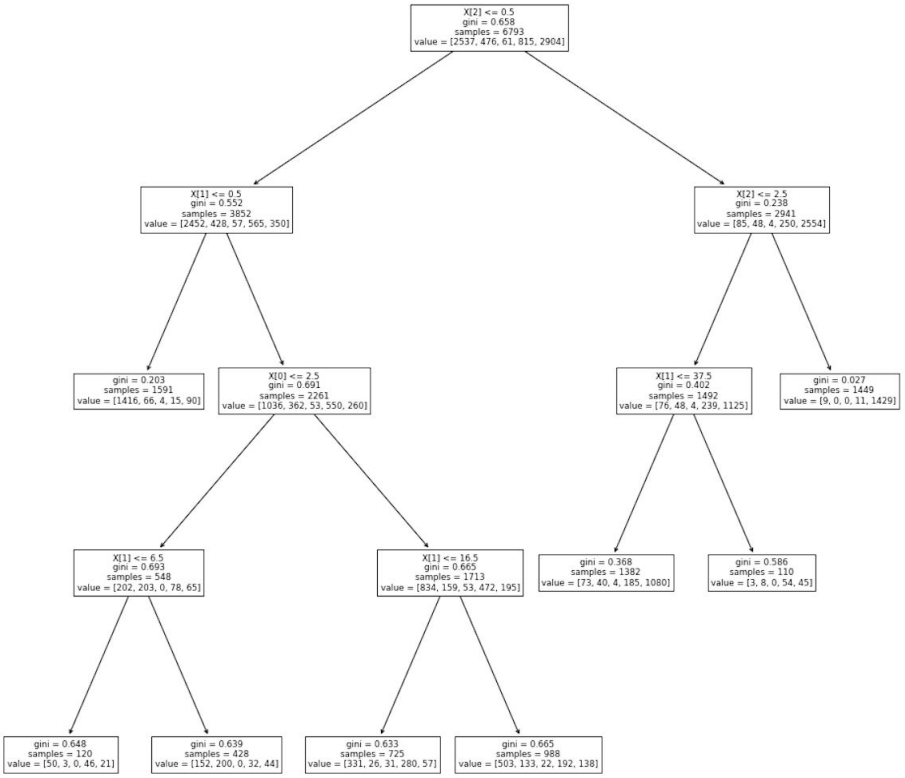


Fig: Arbre de décision pour nb_noeuds=8 et profondeur=8 pour pourcentage de données test=30%

Comme on peut voir sur ce schéma, les produits ont été classifiés en fonction des valeurs de chaque caractéristique. Ainsi, pour le noeud racine, on a classifié selon le nombre d'additifs en lui attribuant une valeur seuil. On fait le même procédé jusqu'à ce qu'on arrive à obtenir une bonne classification de produits qui dépend aussi de la valeur de l'indice de GINI. Ainsi, une valeur d'indice de Gini petite signifie que les produits sont en grande majorité dans une des classes.

On a fait afficher un autre arbre mais en faisant varier le nombre de noeuds et la profondeur de l'arbre en les faisant passer à 25.

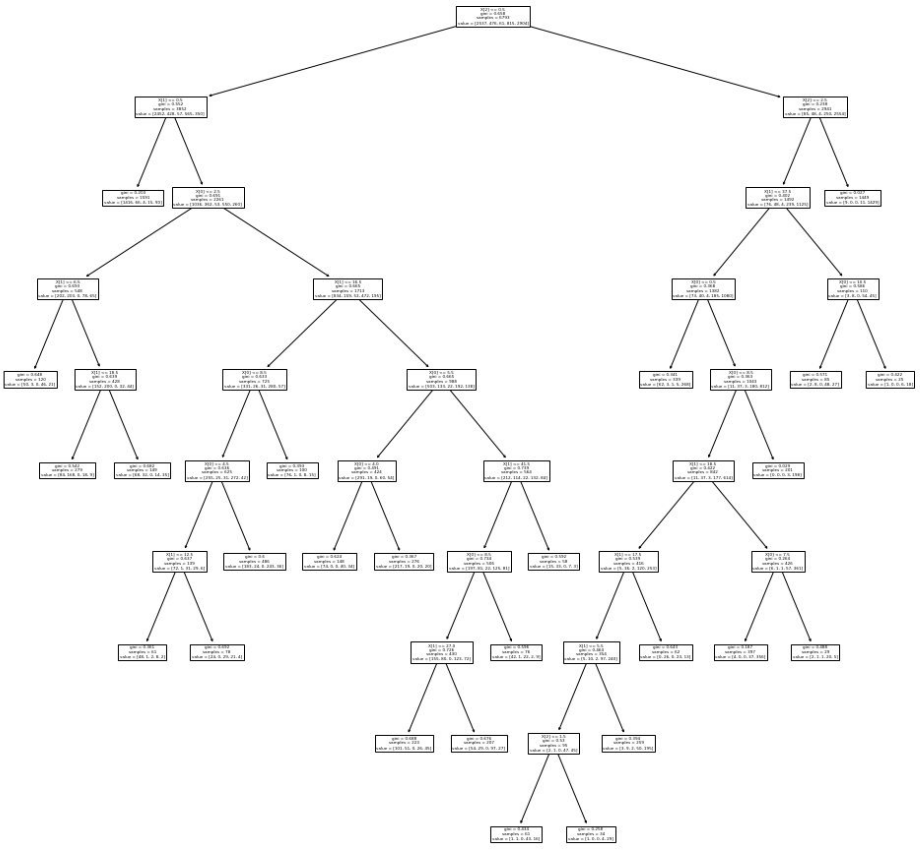


Fig: Arbre de décision pour nb_noeuds=25 et profondeur=25 pour pourcentage de données test=40%

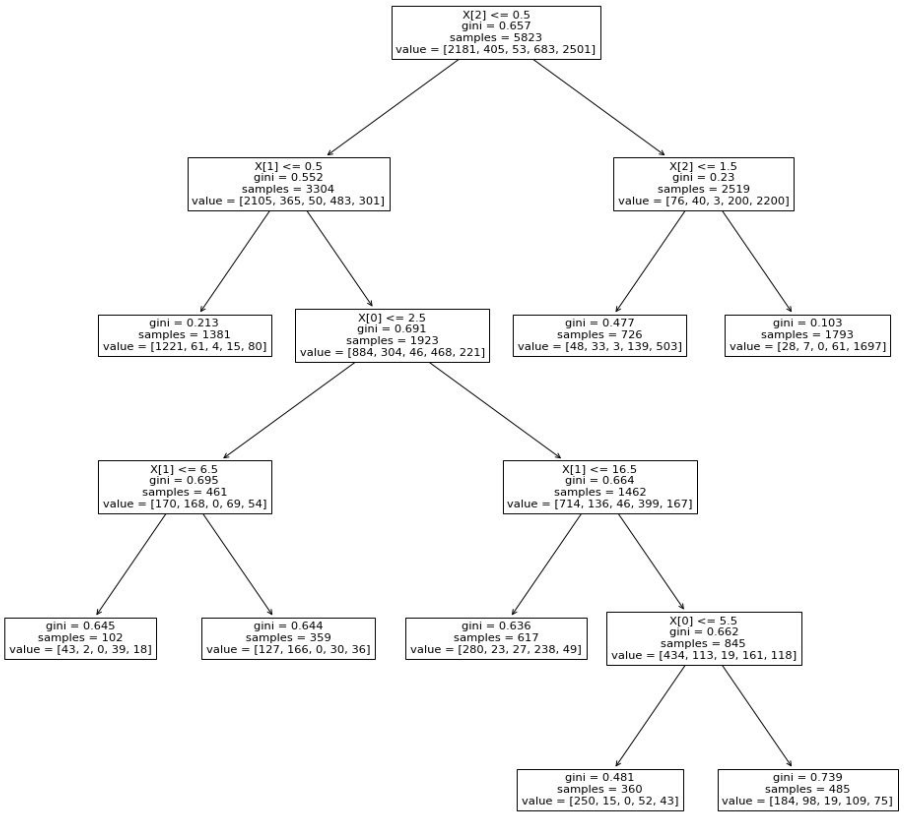


Fig: Arbre de décision pour nb_noeuds=8 et profondeur=8 pour pourcentage de données test=40%

En comparant cet arbre avec l'autre arbre de même paramètre, on peut observer qu'il n'y a pas de différences importantes. On a les mêmes résultats avec un écart négligeable.

Pour évaluer notre modèle de classificateur NOVA, on a décidé de choisir un indicateur élémentaire: l'accuracy qui calcule le pourcentage de données bien prédites. Plus l'accuracy est forte, plus le modèle est robuste aux données à prédire et dispose d'une meilleure performance.

Les résultats obtenus associés à chaque arbre sont les suivants:

Pourcentage de données test=30%
Pour nb_noeuds=8 et profondeur_arbre=8:
-Accuracy sur classification des données d'apprentissage/Test: 0,745

Pour nb_noeuds=25 et profondeur_arbre=25:(voir l'image ci-dessous)
-Accuracy sur classification des données d'apprentissage/Test: 0,779

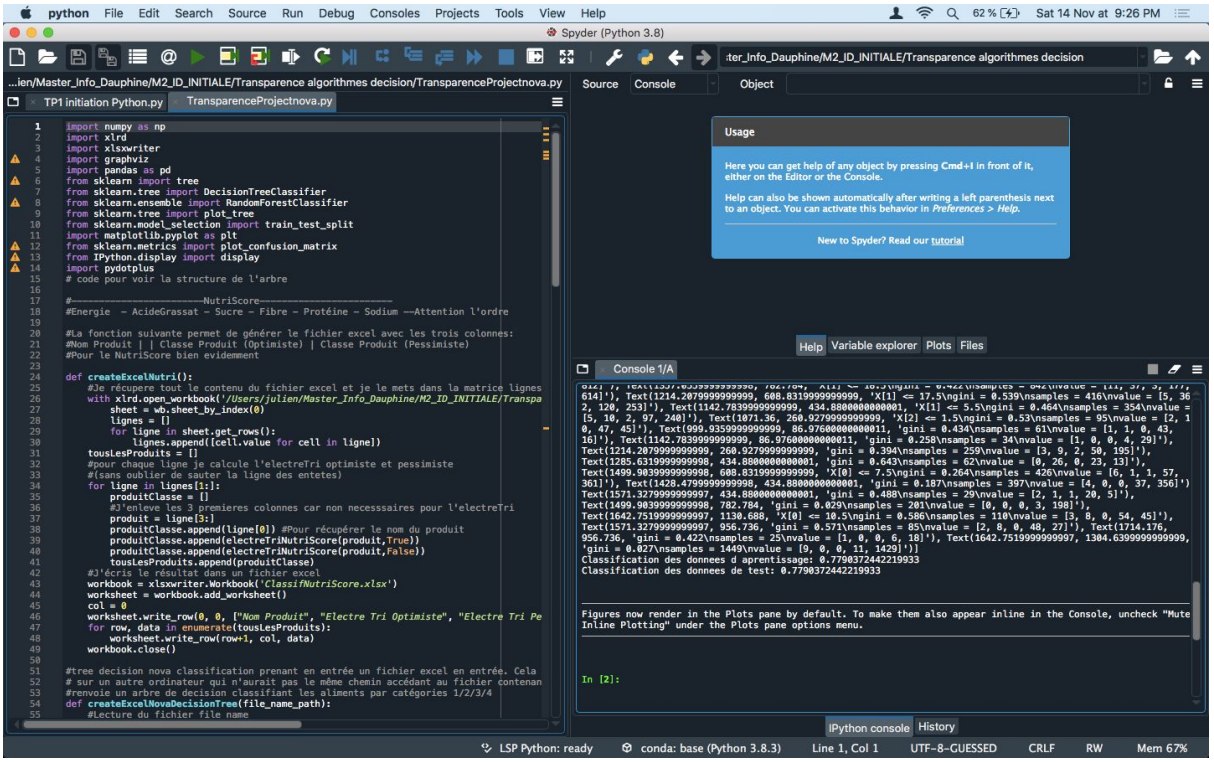


Fig: Exemple d'affichage de l'accuracy pour nb_noeuds=25 et profondeur=25 pour 30% de données à prédire

Maintenant , on va changer le pourcentage de données test en le mettant à 40% au lieu de 30%.

Les résultats obtenus associés à chaque arbre sont les suivants:

Pourcentage de données test=40%
Pour nb_noeuds=8 et profondeur_arbre=8:
-Accuracy sur classification des données d'apprentissage/Test: 0,746

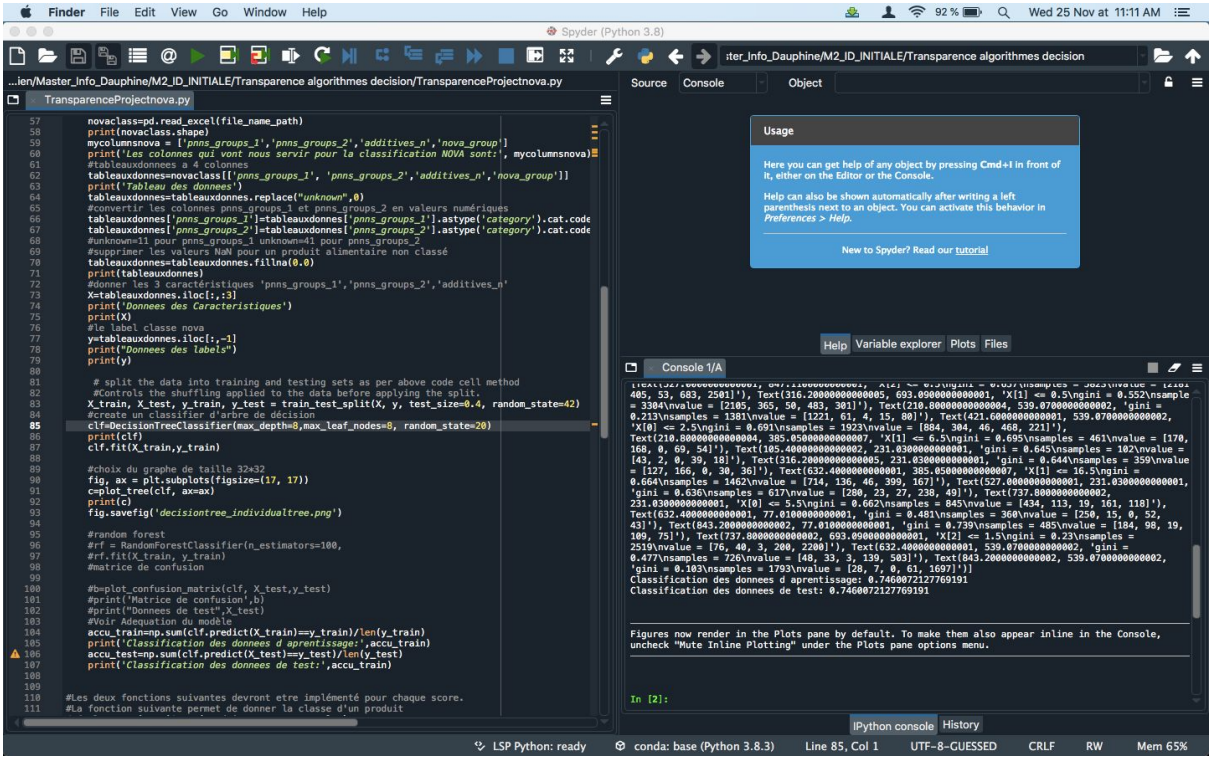


Fig: Exemple d'affichage de l'accuracy pour nb_noeuds=25 et profondeur=25 pour 40 % de données à prédire

Pour nb_noeuds=25 et profondeur_arbre=25:
-Accuracy sur classification des données d'apprentissage/Test: 0,774

D'aprés les résultats, on constate que l'accuracy est sensible au nombre de noeuds et à la profondeur de l'arbre. On aura une meilleure performance lorsqu'on fera augmenter ces deux paramètres. En revanche, le pourcentage de données tests n'a pas un effet important sur l'accuracy. On note des écarts de valeurs au millième près.

On peut en déduire que les arbres de décisions sont un modèle adapté à classifier les produits selon la méthode NOVA.

D: MODÈLE YUKA

Définition:

Yuka est l'application phare de l'agroalimentaire, elle attribue à chaque produit alimentaire une note de 0 à 100.

Le tableau ci-dessous définit l'approche de classification YUKA:

Score	Catégorie
0 à 24	Mauvais
25 à 49	Médiocre
50 à 74	Bon
75 à 100	Excellent

D'après le site internet de YUKA, le calcul de score se base sur 3 dimensions, qui sont:

1. Le Nutriscore
2. Les additifs
3. La dimension BIO

Etant donné que l'algorithme employé pour la classification des aliments demeure inconnu du grand public, notre approche s'est donc fortement inspiré du site internet www.quoidansmonassiette.fr, à noter que ce site appartient au français **Thibault Fiolet,** qui est un chercheur en agro-alimentaire à Paris-Saclay ayant examiné nombreuses méthodes de notation de produits, dont **Yuka**.

Mapping Nutriscore-YUKA:

60% du score YUKA est relatif au score du Nutriscore, c'est-à-dire une correspondance est effectué à partir du Nutriscore du produit pour déduire une note allant de 30 à 90, qui compose un peu moins de deux tiers du score YUKA.

Il est alors nécessaire de noter que dans cette méthode, certains scores Nutriscore ne peuvent pas être mappés, par exemple un produit ayant comme Nutriscore entre -15 et -10 n'a pas un équivalent en terme YUKA.

Ci-dessous un tableau illustrant l'approche de l'étape 1.

Catégorie Nutriscore	Score du Nutriscore	Score YUKA
A	-9 à 7	90
	-6	90
	-4	90
	-3	90
	-2	90
	-1	81
B	0	81
	1	78
	2	75
C	3	69
	4	69
	5	63
	6	60
	7	60
	8	54
	9	51
	10	48
D	11	39
	15 à 19	33
E	20 à 40	30

Pénalisation sur les additifs:

30% du score YUKA est relatif à la pénalisation des additifs, c’est-à-dire, le score de yuka est susceptible de décroitre dans le cas ou le produit comprends des additifs à risque, en effet, YUKA classe les additifs en 4 catégorie : additif sans risque, additif à risque limité, modéré et élevé, comme l'illustre ce tableau ci-dessous:

Type d’additifs	Exemples d’additifs
Additifs sans risque	e261, e296, e322, e325, e330, e334, e420; e421, e422
Additifs à risque limité	e170,e627,e955,e967,e1200
Additifs à risque modéré	e202,e338,e407,e471,e472e,e481,e340ii" ,e341 iii,e223
Additifs à risque élevé	e133,e150d,e250,e252,e450,e450i",e451, e473,e621,e950,e951

De ce fait, YUKA pénalise le score des produits suivant le nombre et la catégorie d’additif que celui-ci contient:

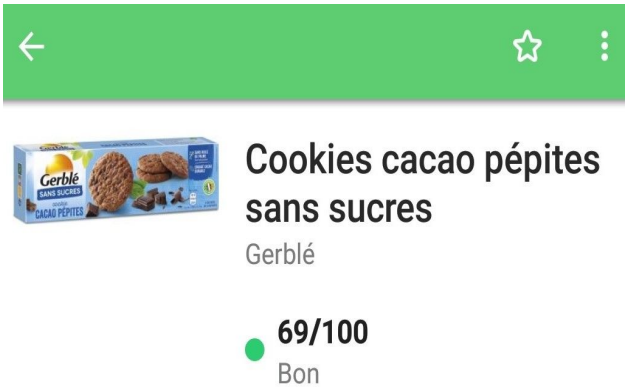
Additifs	Penalisation
Sans risque	Aucune
Risque limité	6
2 x Risque limité	12
Risque limité + risque modéré	24
Risque élevé	30

Bonus relatif à la dimension BIO:

10% de la note est relatif à la dimension Bio, en effet, dans le cas où un produit comprend une étiquette BIO, il se voit ajouter 10 points dans le score YUKA.

Exemple d'application du calcul du score YUKA suivant la méthode ci-dessus:

Produit étudié: Cookies cacao pépites sans sucres



Etape 1: Calcul du Nutriscore et mapping Nutriscore-Yuka

Nutriscore(produit) = (5 + 3 + 0 + 4) - (4 + 4) = 2 Catégorie Nutriscore: **B**

Remarque: Les étapes de calcul du Nutri Score sont mentionnées dans l'énoncé du projet.

Mapping	Note Nutriscore	Note YUKA
	2	75

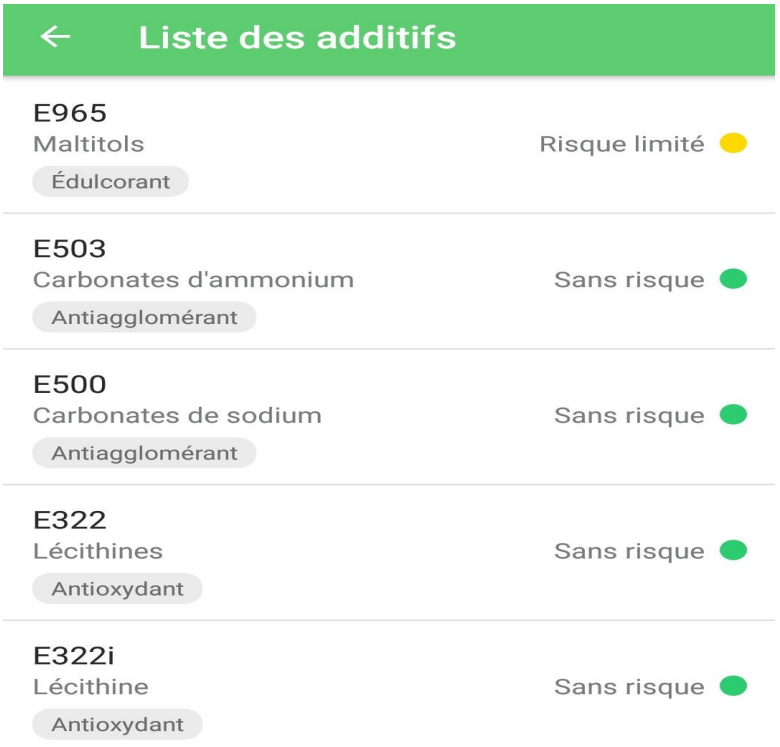
La note **YUKA** à la première étape est donc égale à: **75**

Ci-dessous, la liste aliments de ce produits:

Etape 2: Pénalisation sur les additifs

Ce produit contient 4 additifs:

e965: **Risque limité**
e503, e500, e322, e322i: Sans risque



Dans ce cas, la note YUKA se voit pénalisée de 6 points car le produit comprend un additif à risque limité, le score est alors de: 69.

Yuka(Produit) = Yuka(Produit) - 6 = 75 - 6 = 69

Etape 3: Bonus BIO

Ce produit n'est pas étiqueté de BIO, donc la note reste inchangée.

Yuka(produit) = **69/100**

On voit bien que la méthode de calcul qu'on a adoptée correspond bien au score renvoyé par l'application YUKA.

Néanmoins, cette application présente quelques incohérences concernant le calcul du score des produits, dont:

- Les erreurs concernant la composition des produits, notamment la quantité de certains aliments qui sont différents de la réalité.
- Plusieurs produits de qualité nutritionnelle équivalente n'ont pas la même note YUKA
- Incohérence vis-à-vis des additifs, en effet, certains additifs connus pour être dangereux pour la santé ne sont pas inclus dans la liste additifs utilisée pour le calcul de ce score.
- La note YUKA des sodas soulève beaucoup de questionnement, notamment, les sodas sans sucre, ces derniers ont pour la plupart une note médiocre malgré que la logique YUKA les classifie parmi les aliments bons.

II: Construction des bases de données BD2, BD3

Remarque: La génération de cette base de données est effectuée à l'aide d'un script Python qu'on vous joins. Pour la construction de cette base de données, on s'est basé sur le fichier excel fourni ‘’openfoodfacts_simplified_database.xlsx".

A - BD2

Cette base de données comprend 1220 produits ayant pour attributs la note et la catégorie Nutriscore, ainsi que la classe Nova. Chaque catégorie du Nutriscore est représentée par plus de 200 produits.

B-BD3

Cette base de données comprend 75 produits étiquetés de sa catégorie Nutriscore, sa classe Nova et YUKA, ainsi que le nombre d'additifs qu'il contient, une colonne BIO est présente pour indiquer si celui-ci est BIO. Chaque catégorie du Nutriscore est représentée par 15 produits.

Dans le but de maximiser les chances d'avoir une intersection de moins de 20% avec les BD des autres groupes, on sélectionne à chaque itération 3 score du Nutriscore qui ne seront pas acceptés dans nos Base de données.

A noter que le score YUKA est calculé suivant la méthode présentée dans ce rapport.

III: Interface graphique

1. Architecture technique

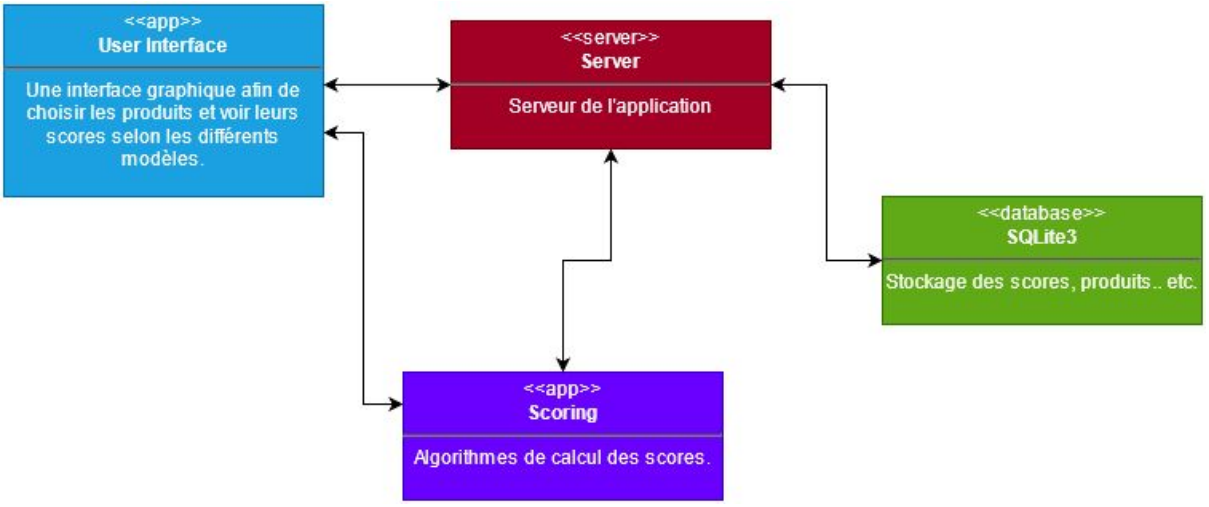


Figure - Architecture du système

Notre interface graphique est alimentée par une base de données contenant les informations sur chaque produit:

foodscore > frontend > models.py > ...

```python
from django.db import models

# Create your models here.
class Produit(models.Model):
    name = models.CharField(max_length=255)
    ns_electretri_opti = models.CharField(max_length=30)
    ns_electretri_pess = models.CharField(max_length=30)
    nova = models.CharField(max_length=30)
    yuka = models.CharField(max_length=30)
    energie = models.CharField(max_length=30)
    gras = models.CharField(max_length=30)
    sucre = models.CharField(max_length=30)
    fibre = models.CharField(max_length=30)
    proteine = models.CharField(max_length=30)
    sodium = models.CharField(max_length=30)

```

Les scores sont calculés à travers nos différents algorithmes de calcul de scores et sont ensuite stockés dans la base de données du serveur de l'interface graphique.

2. Méthode de déploiement

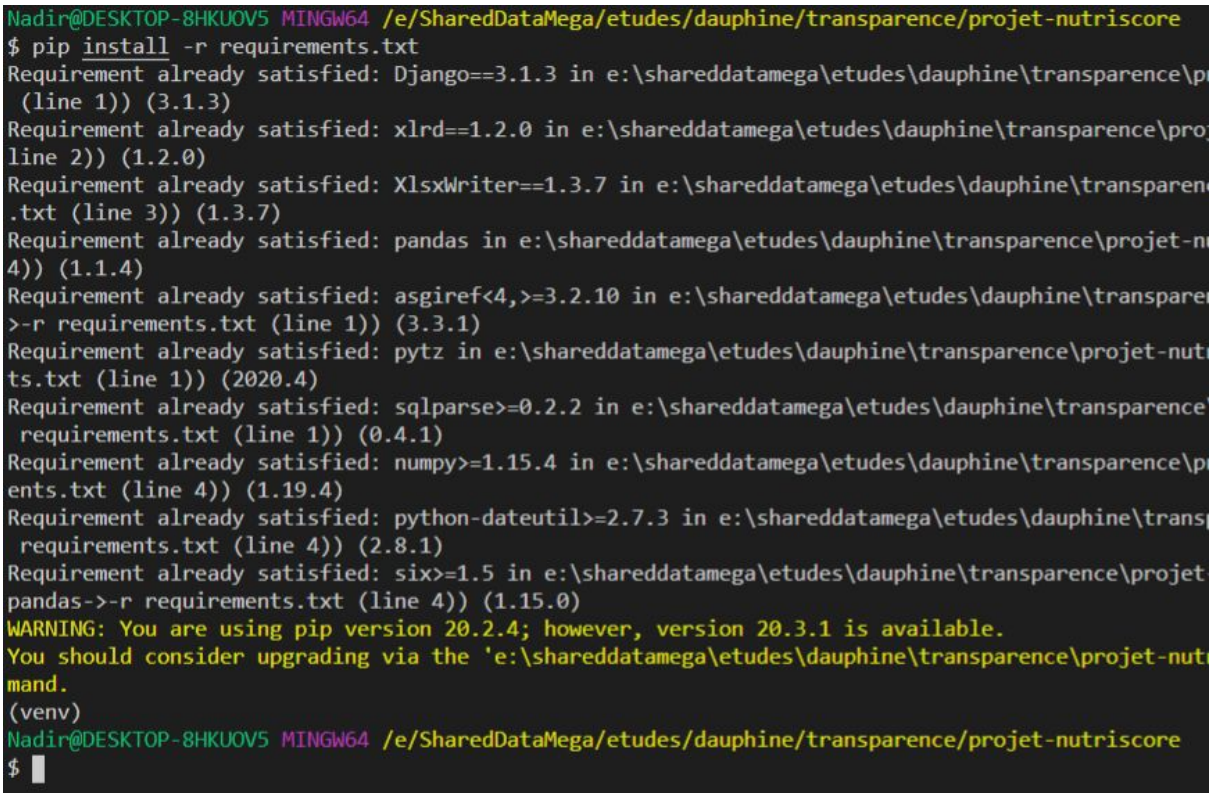
Les fichiers du serveur se retrouvent dans le zip du projet.
Le déploiement est fait de la façon suivant:

- Il faut avoir python 3 installé sur la machine.
-

Naviguer à travers une console vers le dossier **projet-nutriscore**.

Installer les requirements python pour le serveur:

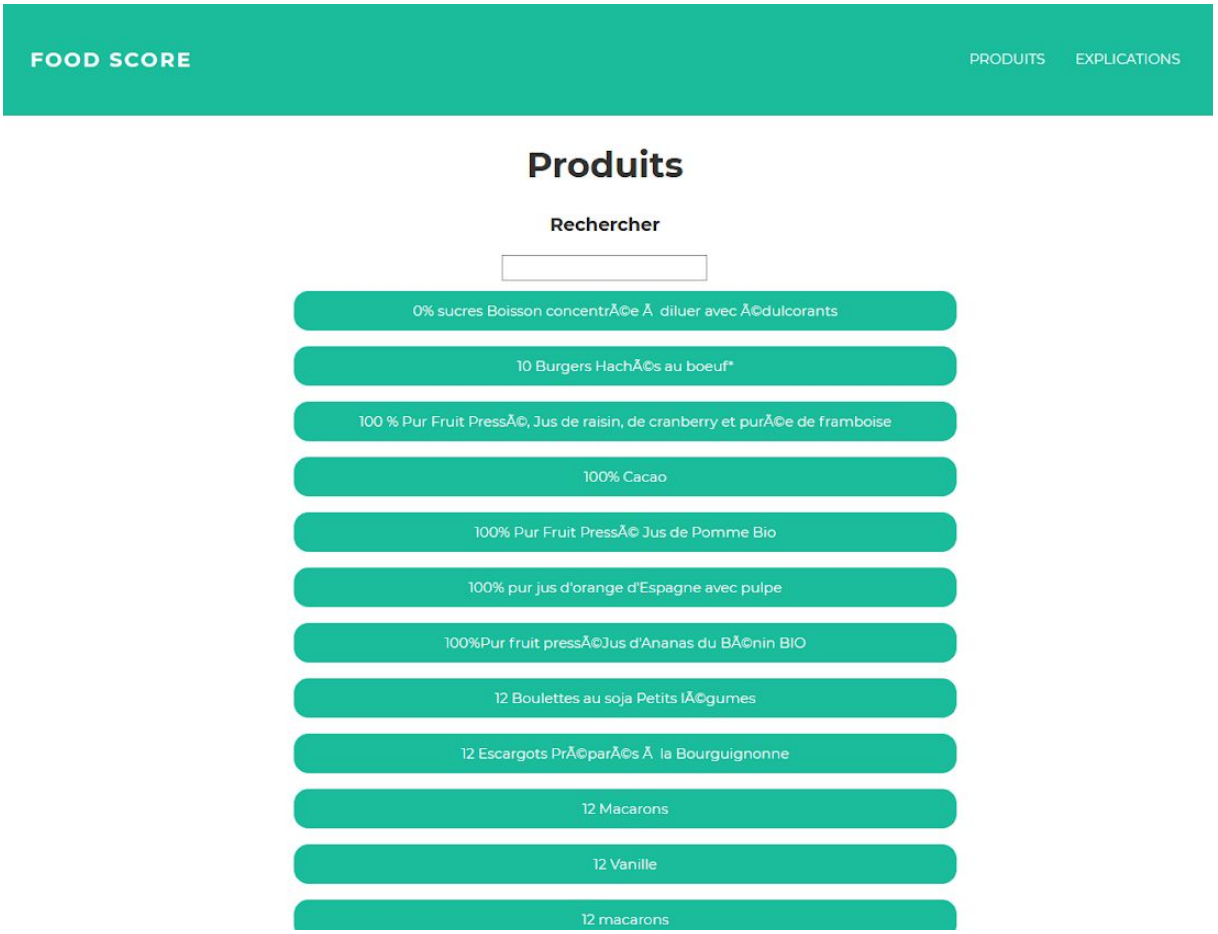
 # Il faut s'assurer d'utiliser python3 et non pas python2. Il est possible que la commande à utiliser sur la machine soit pip3 au lieu de pip.



Une fois que les packages sont installés, lancer le serveur:
 # Il faut s'assurer d'utiliser python3 et non pas python2. Il est possible que la commande à utiliser sur la machine soit python3 au lieu de python.

L'interface sera accessible sur l'adresse 127.0.0.1:8000

3. Aperçu de l'interface



Produits

Rechercher

0% sucres Boisson concentrÃ©e Ã diluer avec Ã©dulcorants

10 Burgers HachÃ©s au boeuf*

100 % Pur Fruit PressÃ©, Jus de raisin, de cranberry et purÃ©e de framboise

Nutriscore optimiste

E

Nutriscore pessimiste

E

Yuka

54

Nova

1

En savoir plus sur comment les scores sont calculés

Energie

241.0

Gras

0.0

Sucre

12.9

Fibre

0.1

Proteine

0.6

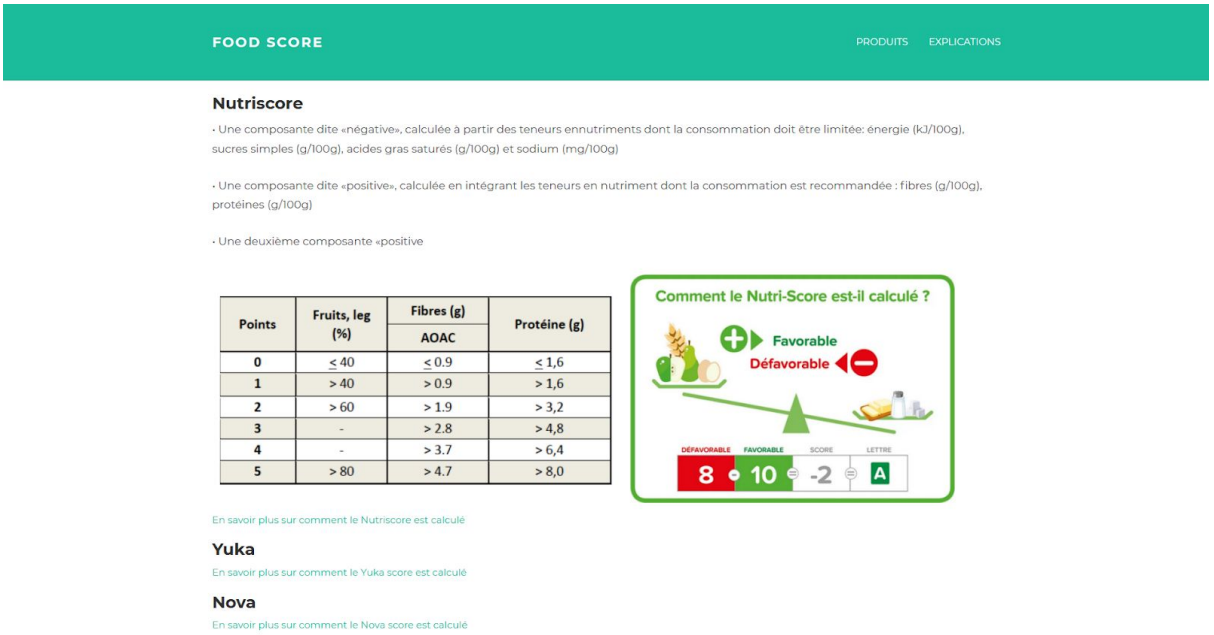
Sodium

0.012

100% Cacao

100% Pur Fruit PressÃ© Jus de Pomme Bio

100% pur jus d'orange d'Espagne avec pulpe



4. Démonstration

Avec les fichiers du projet sont joints des vidéos montrant l'interface graphique.

IV: Etude des propriétés de transparence de chaque modèle

1. **Loyauté**

   ✅ Notre solution respecte le critère de loyauté. En effet, elle affiche à l'utilisateur le même fonctionnement connu par le fournisseur (nous même), qui est le calcul et l'affichage de scores.

2. **Équité**

   ✅ Le notre solution est conforme à l'équité car les résultats qu'il produit n'induisent pas un effet discriminant ou biais à l'égard d'une catégorie particulière de la population.

3. **Responsabilité**

   ✅ Notre solution permet de justifier les résultats en montrant les aliments contenus dans chaque produit qui ont mené au score en question. Il est conforme au critère de responsabilité.

4. **Explicabilité**

   ✅ Les algorithmes sont expliqués aux utilisateurs leur permettant de comprendre pourquoi un produit a eu un score en particulier avec une méthode spécifique.

5. **Intelligibilité**

   ✅ Notre solution est intelligible car nous avons la capacité de comprendre et vérifier les résultats obtenus ainsi que leur précision.