PROJET 2

Concevez une application

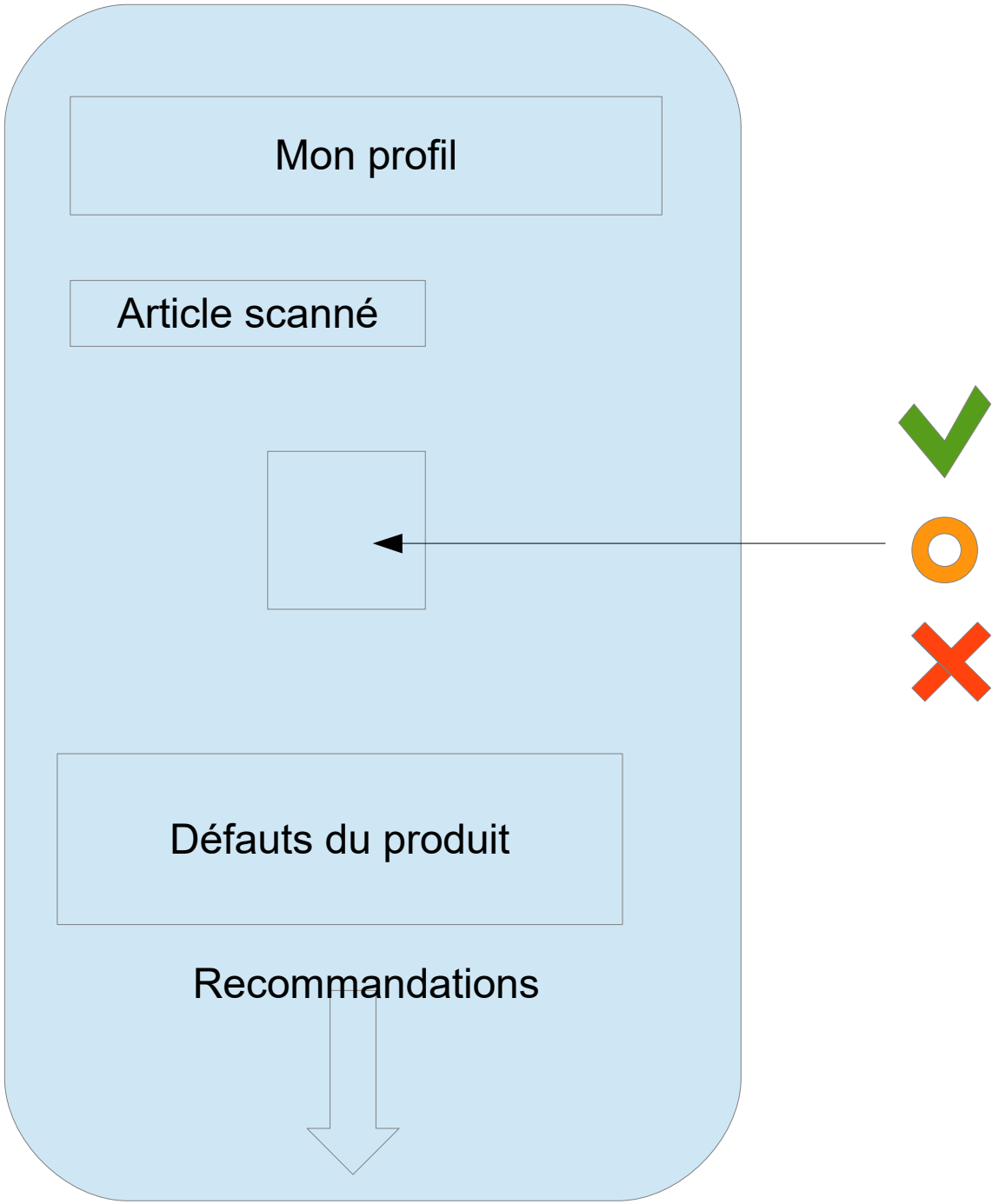
au service de la santé publique

Plan

- Présentation de l'application
- Nettoyage des données
- Analyse des données
- Complétion des données manquantes
- Réduction de dimension

Présentation de l'application

- L'utilisateur défini son profil de consommateur parmi un ensemble de profils prédéfinis.
- L'application lui indique si le produit qu'il scanne est conforme à son profil.
- L'application lui propose de meilleurs produits.


Mon profil
Article scanné
Défauts du produit
Recommandations

Le jeux de données

- Initialement, c'est un tableau comportant environ 320 000 lignes (des produits) et 160 colonnes (des caractéristiques concernant les produits)
- Objectif : trouver une application réalisable à partir de ce jeux de données .
- Problème : les données sont incomplètes et contiennent des erreurs.

Problème :

On ne peut pas imaginer une application seulement à partir de l'intitulé des différentes colonnes.

Exemple : il y a des colonnes indiquant les quantités de vitamines, on pourrais penser à une application suggérant des produits avec beaucoup de vitamines, mais en fait ces colonnes sont quasiment vides.

Les données déterminent les applications possibles, donc on va analyser les données pour voir ce qui est faisable.

On ne va pas analyser des cases vides, donc on commence par nettoyer le jeux de données.

Nettoyage des données

- Suppression des colonnes
- Suppression des lignes
- Traitement des valeurs manifestement fausses

Suppression de colonnes

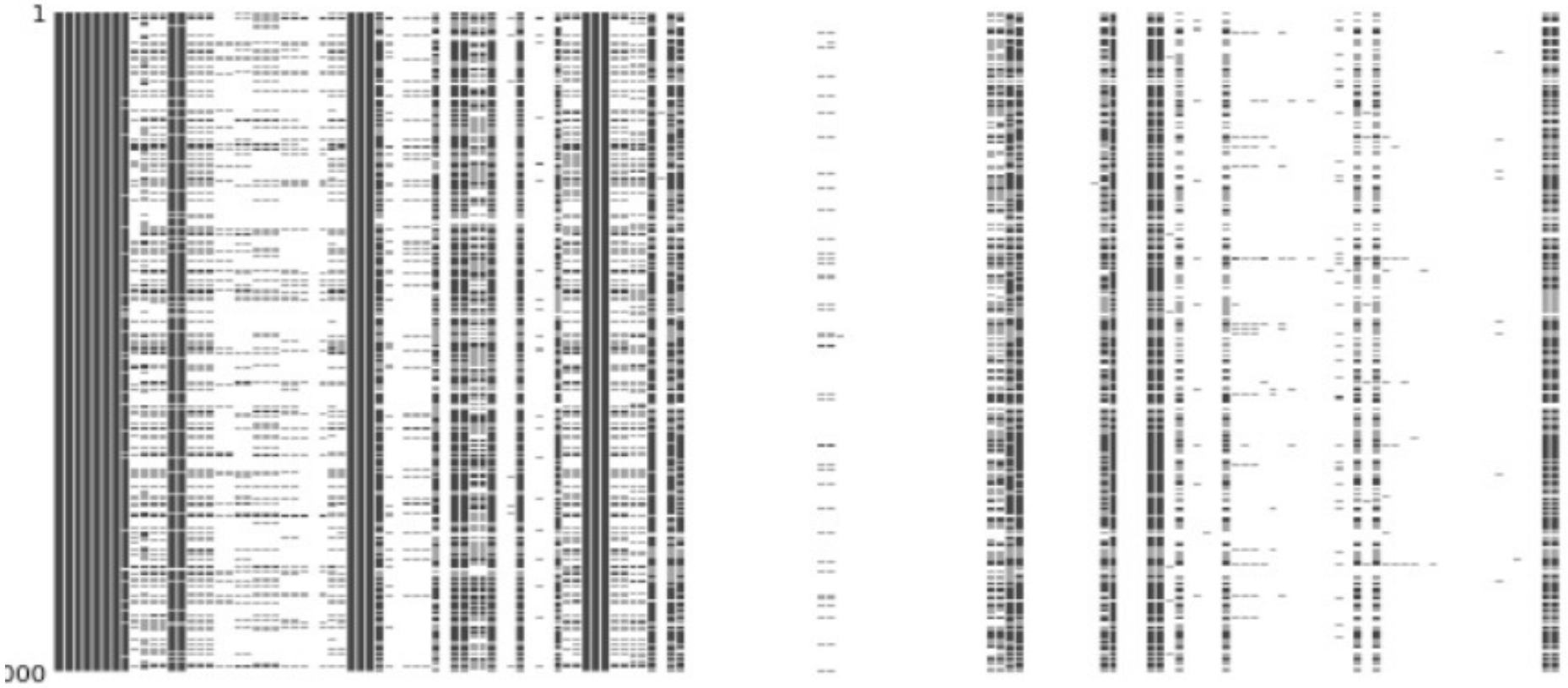
A partir d'un certain seuil de valeurs manquantes, une colonne est supprimée. A l'exception de deux colonnes contenant des catégories de produit, car il semble difficile de se passer des catégories pour faire des recommandations.

De plus certaines informations sont présentes en double comme le pays en anglais et le pays en français. On supprime les doublons.

Suppression de lignes

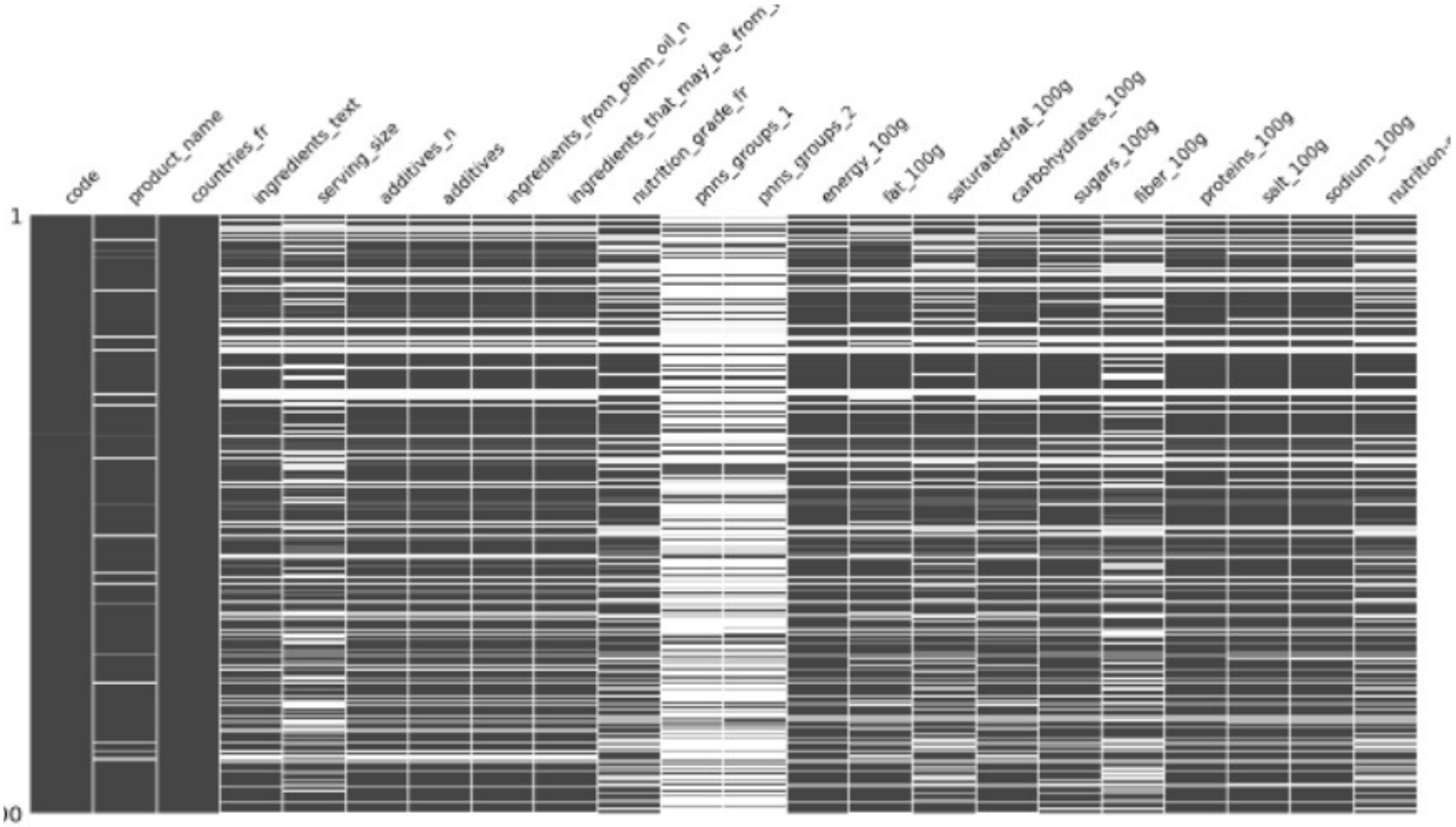
Comme pour les colonnes, à partir d'un certain seuil de valeurs manquantes, une ligne est supprimée.

Valeurs manquantes



Les zones en blanc indiquent les valeurs manquantes.

Valeurs manquantes après avoir supprimé des colonnes



Valeurs aberrantes

On remplace les valeurs manifestement fausses par des Nan.

energy_100g	fat_100g	saturated-fat_100g	carbohydrates_100g	sugars_100g	fiber_100g	proteins_100g	salt_100g	sodium_100g	nutrition-score-fr_100g
231199.0	NaN	550.0	NaN	3520.0	5380.0	430.0	0.01016	0.004	25.0

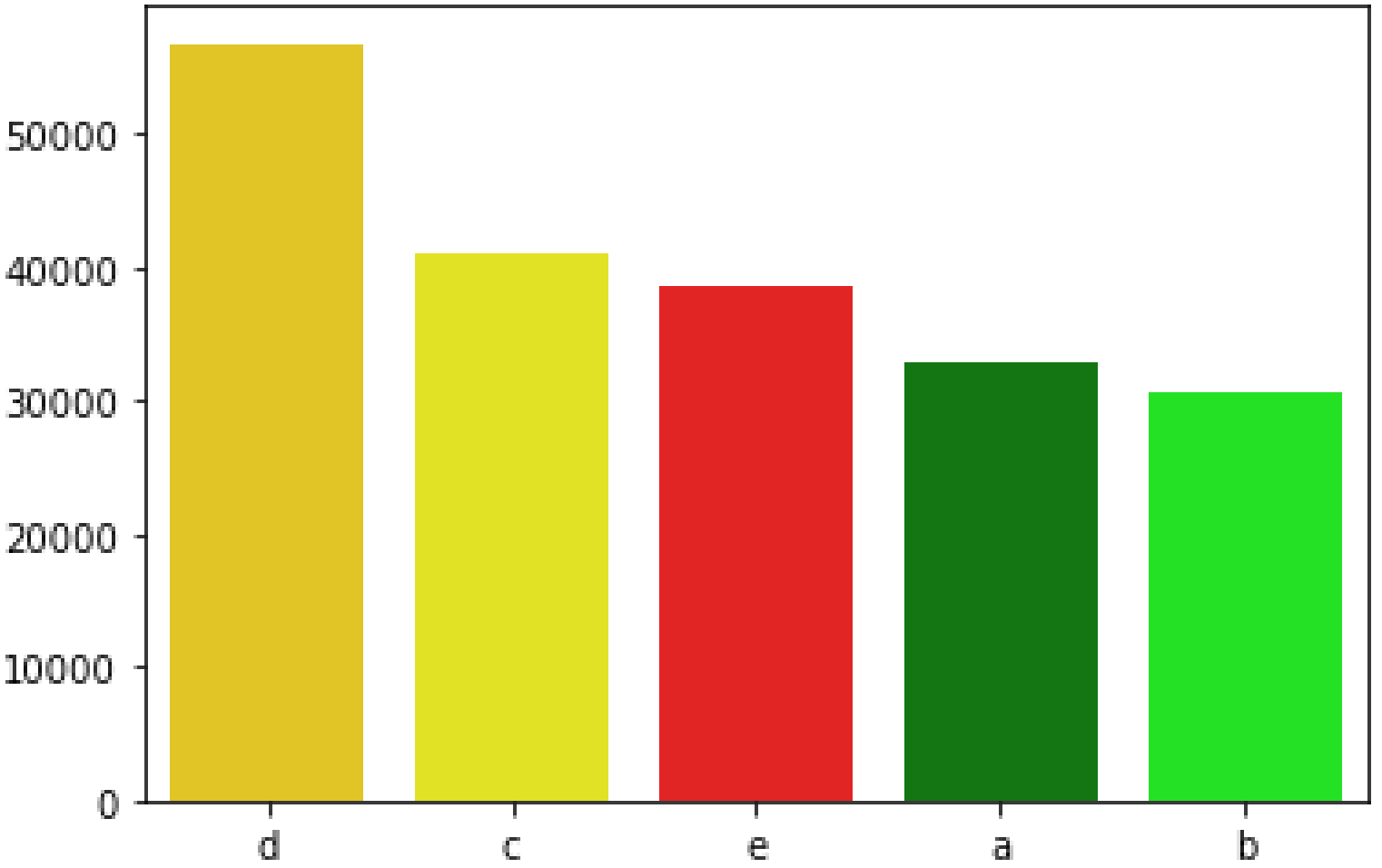
Une masse pour 100g doit être comprise entre 0 et 100g !
Une quantité d'énergie pour 100g doit être comprise entre 0 et 900 kcal.

	nbre_de_nan_avant	nbre_de_nan_après	difference
energy_100g	154	96565	96411
fat_100g	1038	1040	2
saturated-fat_100g	21382	21385	3
carbohydrates_100g	1038	1049	11
sugars_100g	6456	6470	14
fiber_100g	48045	48048	3
proteins_100g	579	582	3
salt_100g	732	800	68
sodium_100g	747	774	27

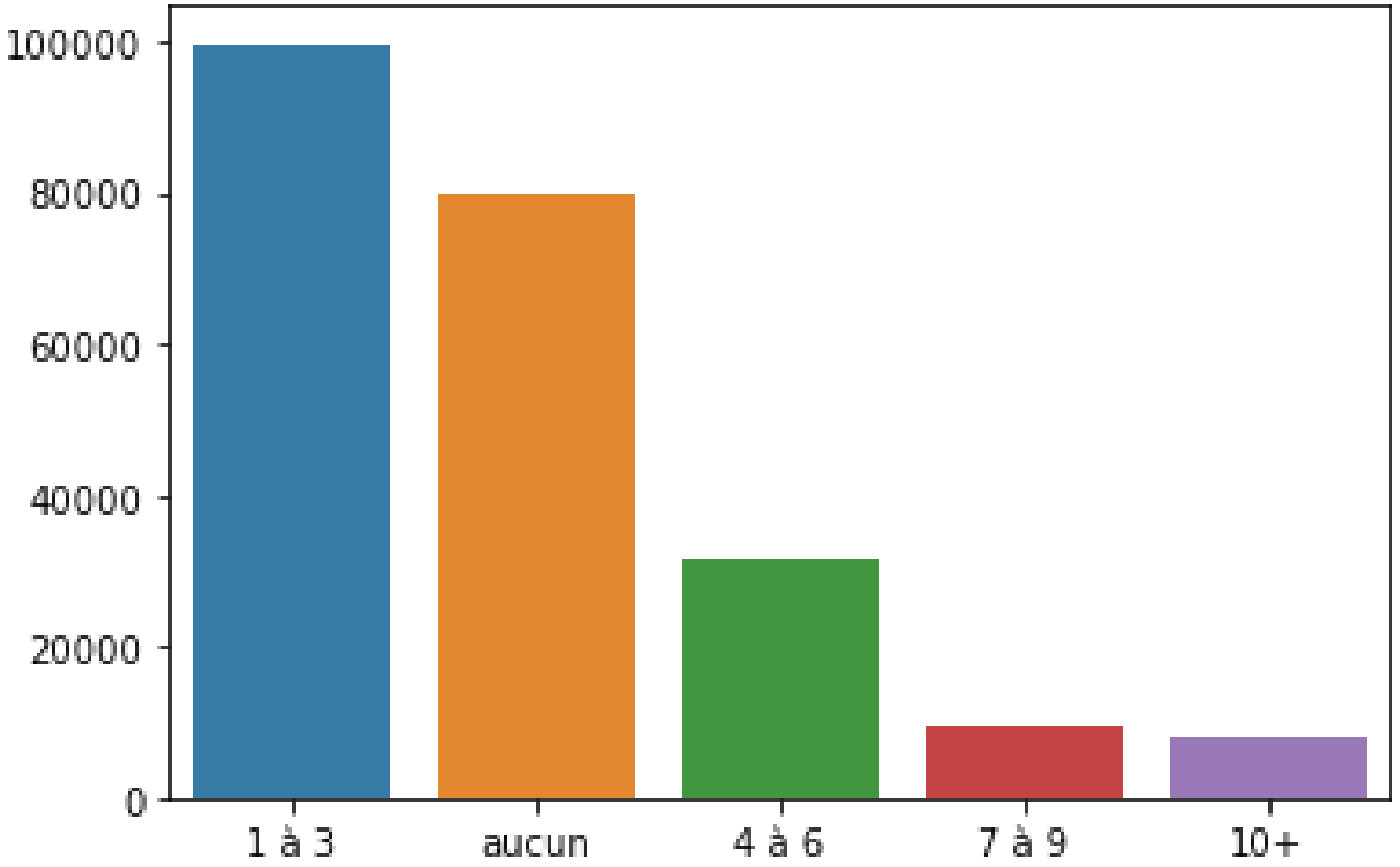
Analyse des données

Objectif : voir ce qui est faisable.

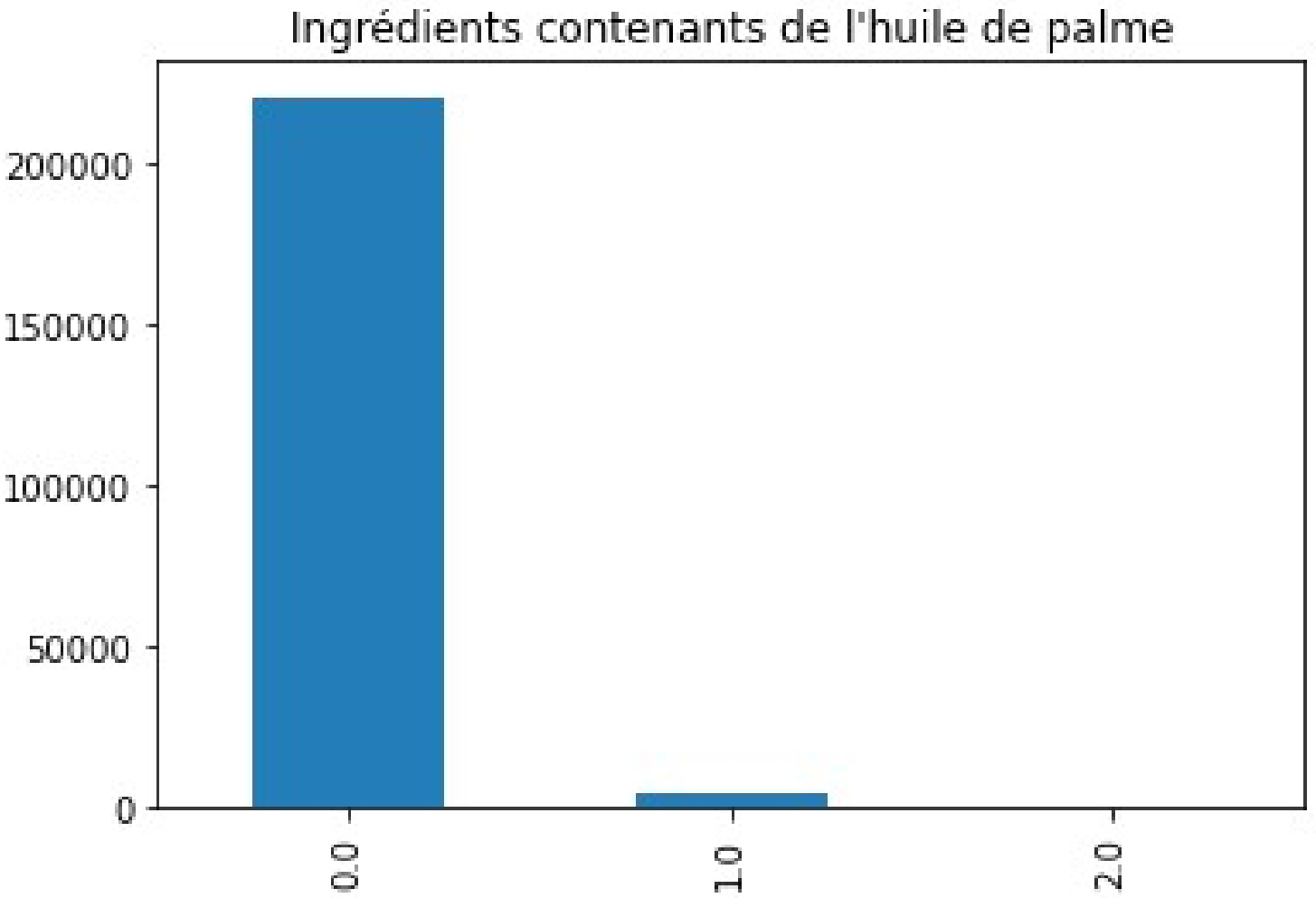
Nutri-Score



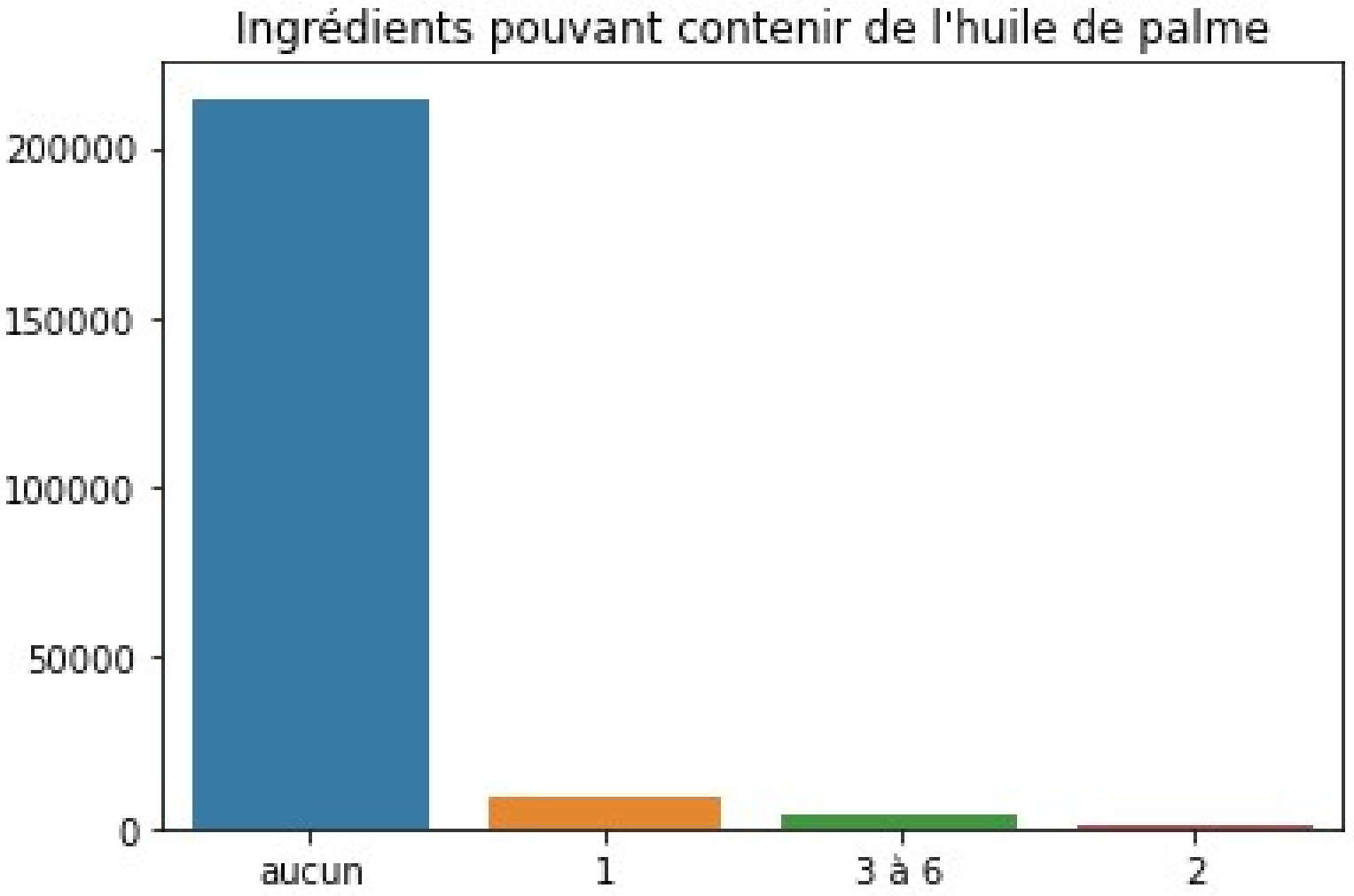
Additifs



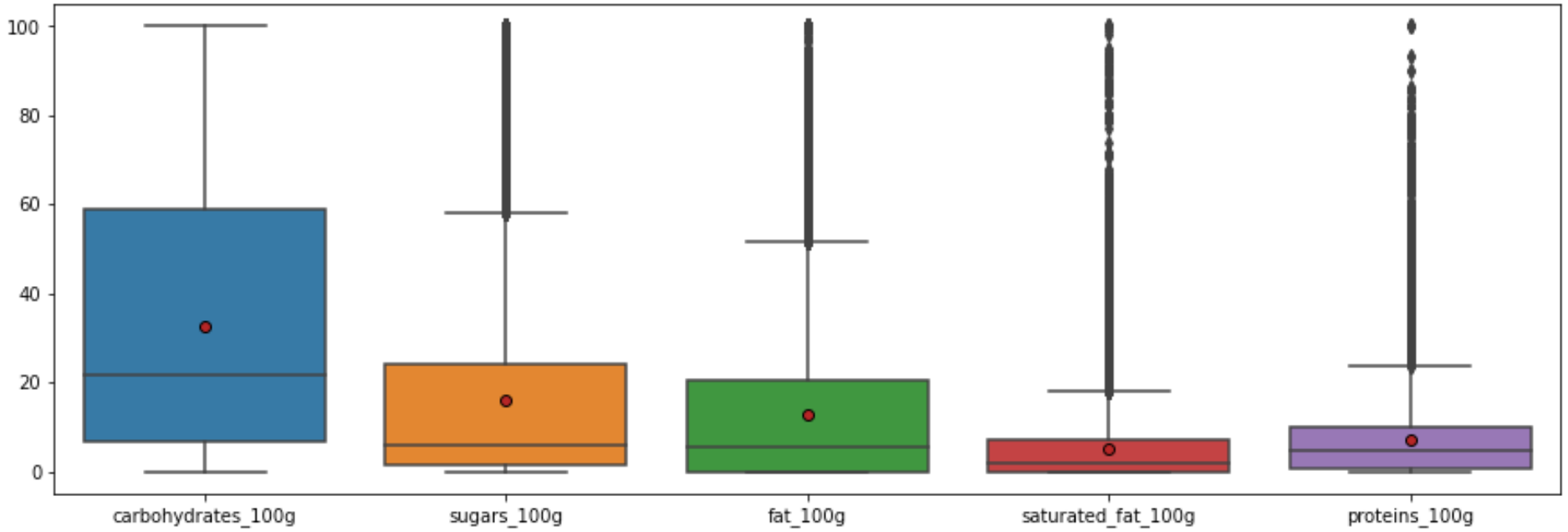
Huile de palme



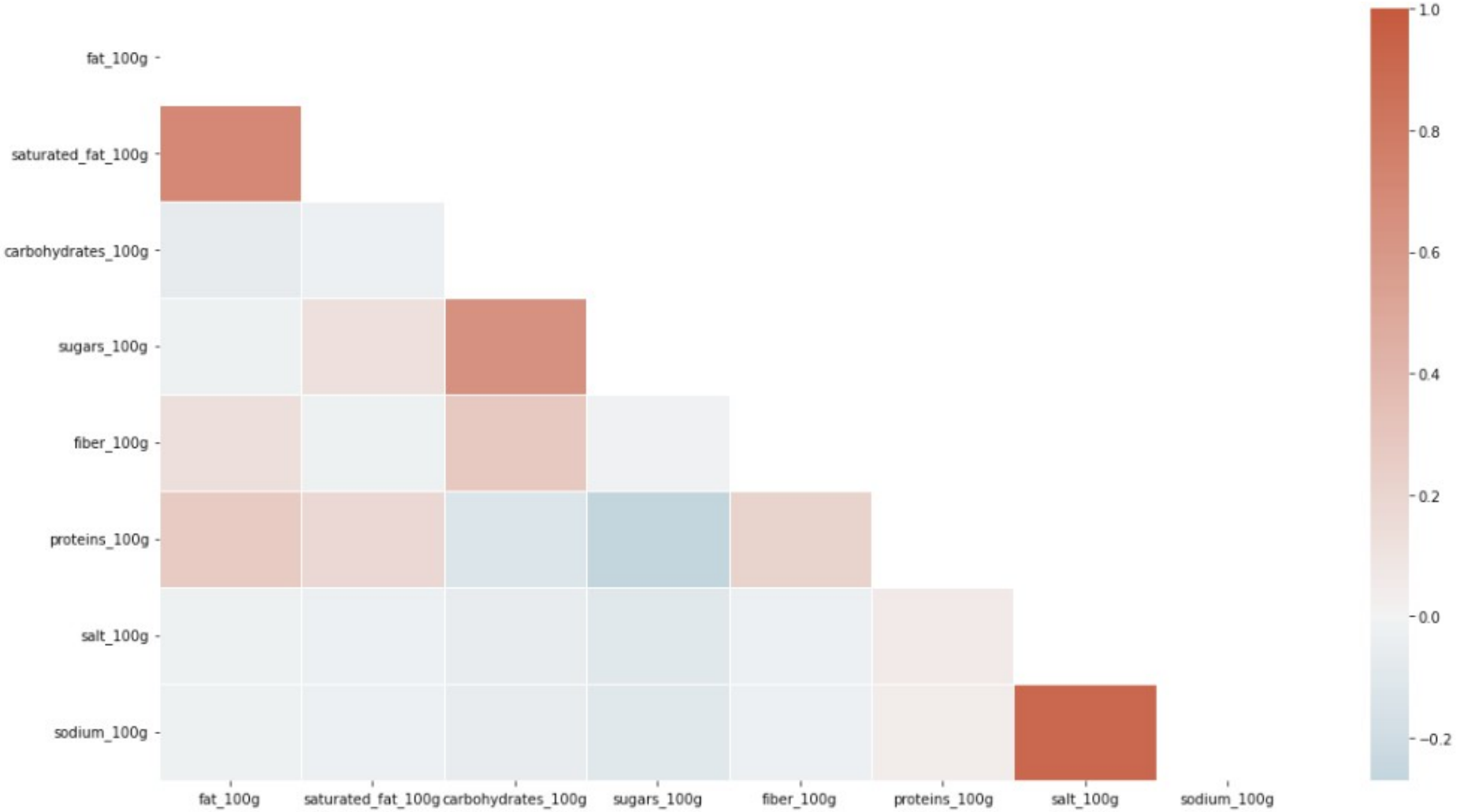
Huile de palme



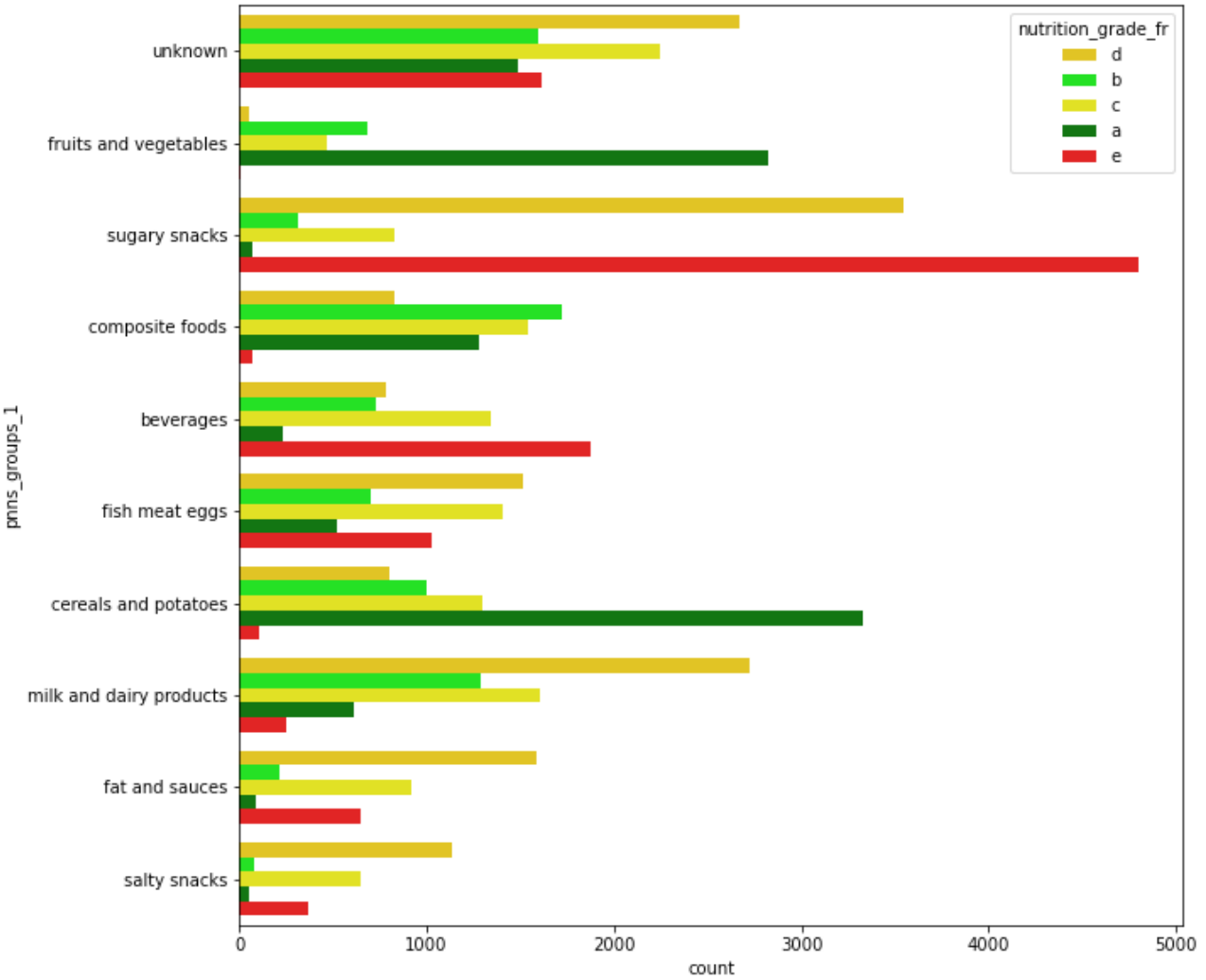
Valeurs nutritionnelles



Corrélation des valeurs nutritionnelles pour 100g



Nutri-Score par catégorie



Pour certaine catégories comme par exemple 'sugary snacks', les recommandation seront limité.

Analyse de la variance en fonction de la catégorie

	eta_carre
fat_100g	0.565177
saturated_fat_100g	0.492992
carbohydrates_100g	0.638666
sugars_100g	0.563007
fiber_100g	0.208751
proteins_100g	0.514014
salt_100g	0.043938
sodium_100g	0.040146

- La catégorie a une influence sur les valeurs nutritionnelles.
- On peut donc envisager de recommander des catégories de produits en fonction du profil de l'utilisateur.

Traitement des valeurs manquantes

- Quantités pour 100g
- Valeur numérique du Nutri-Score
- Nutri-Score (classe)

Traitement des valeurs manquantes

- Quantités pour 100g  Médiane par catégorie
- Valeur numérique du Nutri-Score
- Nutri-Score (classe)

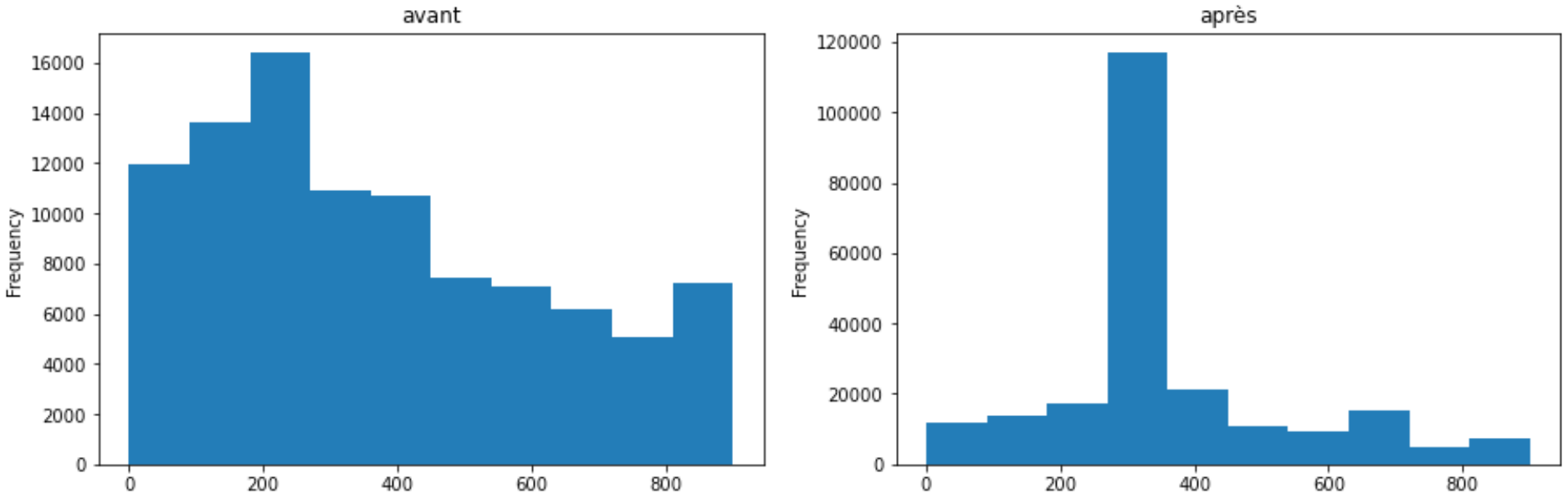
Traitement des valeurs manquantes

- Quantités pour 100g → Médiane par catégorie
- Valeur numérique du Nutri-Score → Régression linéaire
- Nutri-Score (classe)

Traitement des valeurs manquantes

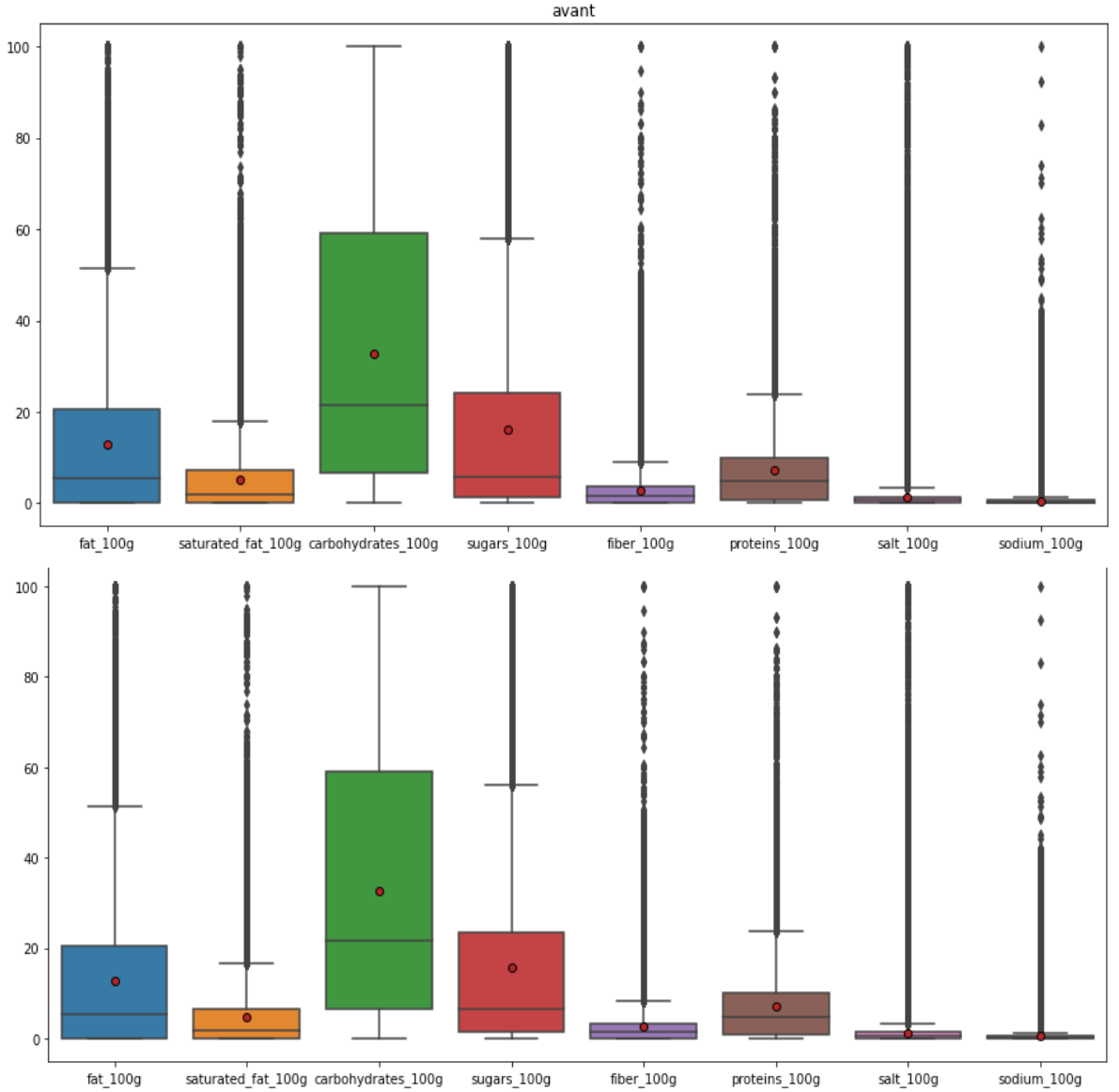
- Quantités pour 100g → Médiane par catégorie
- Valeur numérique du Nutri-Score → Régression linéaire
- Nutri-Score (classe) → k-NN

Ajout des quantités d'énergie manquantes

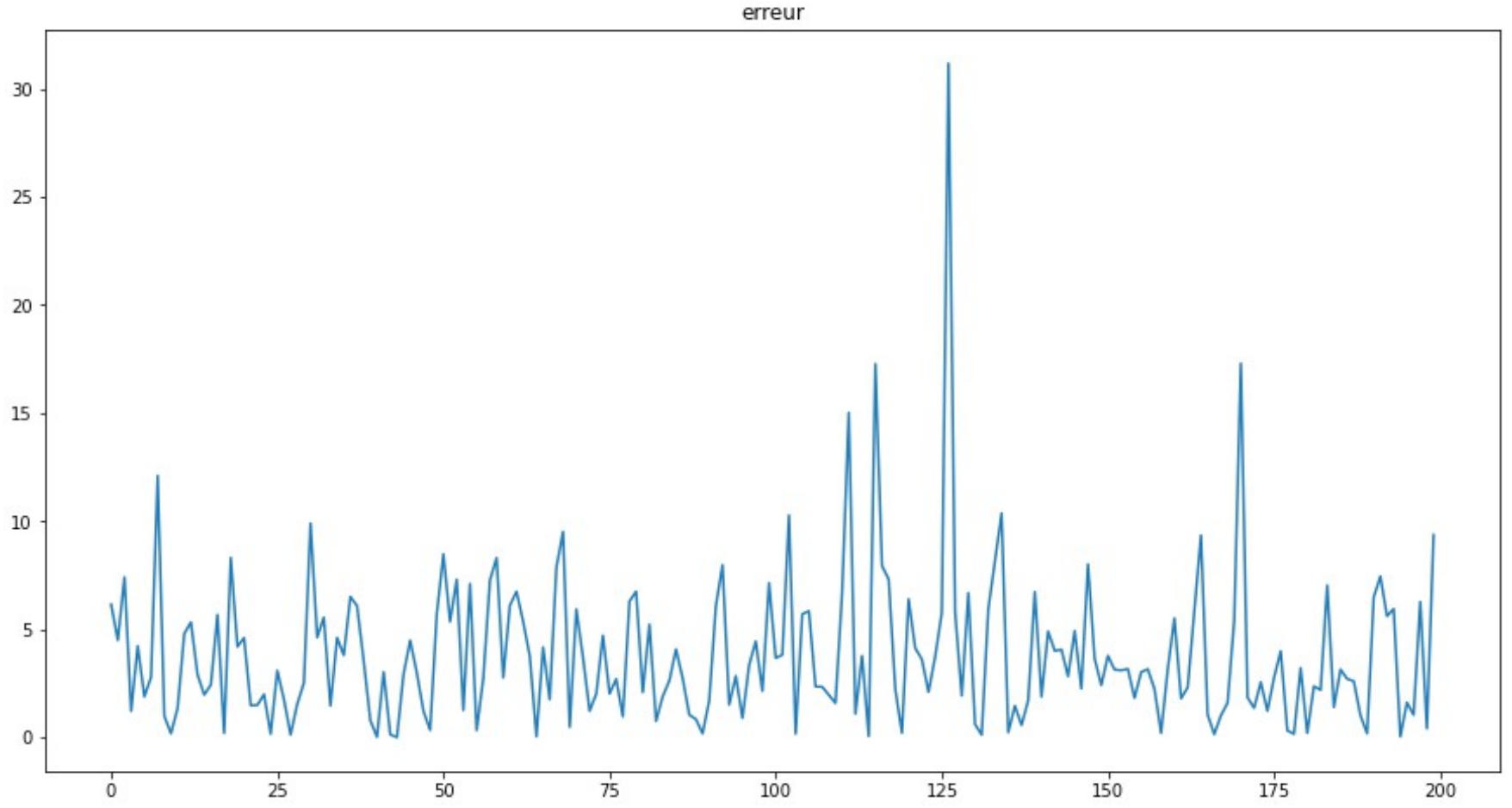


La distribution est très différente, donc cette complétion n'est pas pertinente.

Pour les nutriments on ne voit presque pas de différences, donc la complétion est pertinente.



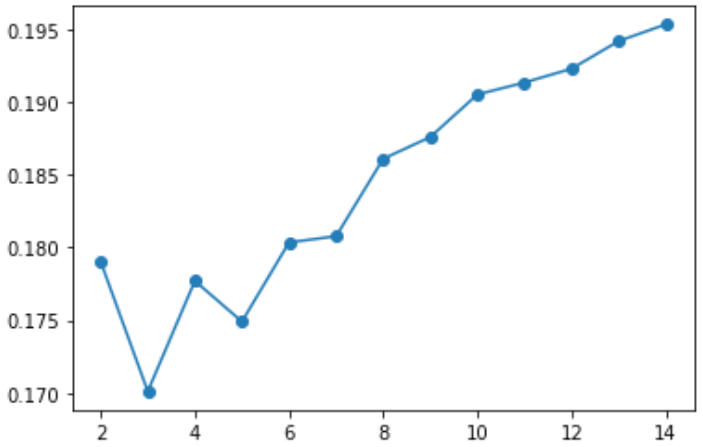
Régression linéaire pour le Nutri-Score



L'erreur quadratique moyenne est proche de 5 alors que les valeurs doivent être comprise entre -15 et 40 et qu'il y a une classe de Nutri-Score ayant une amplitude de seulement 2.

K-NN pour le Nutri-Score grade

```
errors = []
for k in range(2, 15): # on test différentes valeurs pour l'hyperparamètre k
    knn = neighbors.KNeighborsClassifier(k)
    errors.append(1-knn.fit(Xtrain, Ytrain).score(Xtest, Ytest))
plt.plot(range(2,15), errors, 'o-')
plt.show()
```

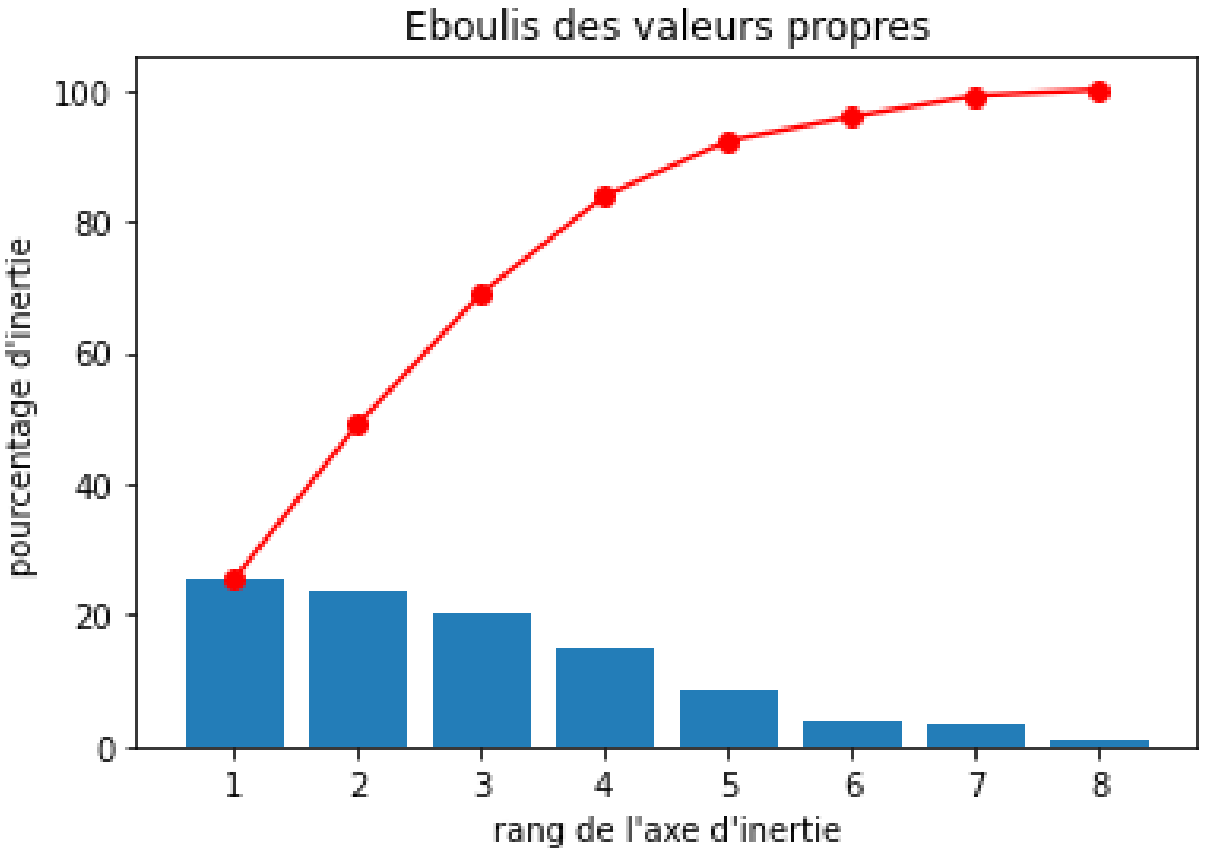



```
# k=3 semble être le meilleur choix
knn = neighbors.KNeighborsClassifier(n_neighbors=3)
knn.fit(Xtrain, Ytrain)
erreur = 1 - knn.score(Xtest, Ytest)
erreur
```

```
0.17010425260631512
```

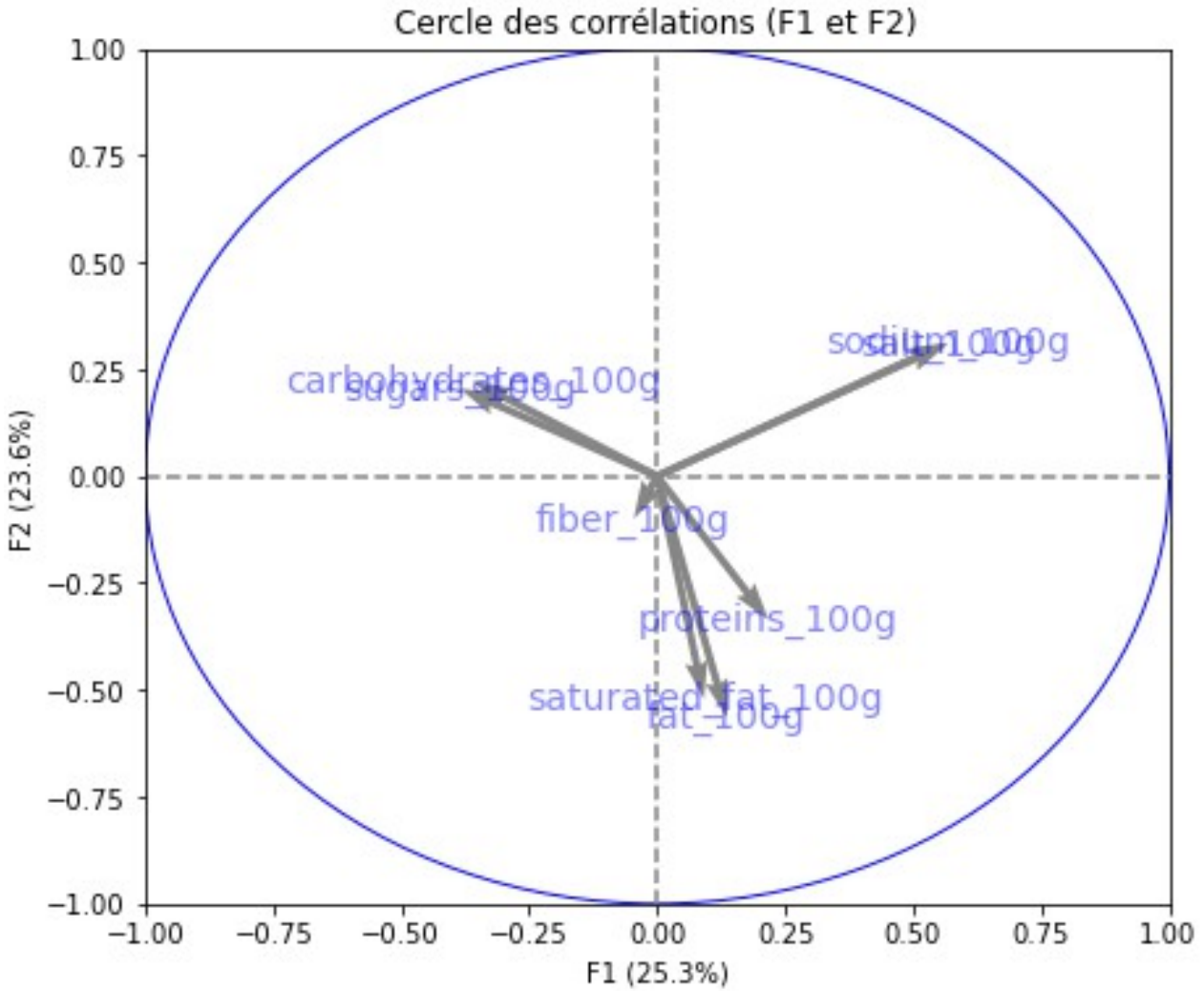
Notre modèle effectue une prédiction correcte sur l'échantillon de test dans plus de 80% des cas.

ACP



Les quatres premières composantes permettent de d'écrire 80% des données.

ACP


Cercle des corrélations (F1 et F2)
F1 (25.3%)
F2 (23.6%)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
fiber_100g
proteins_100g
saturated_fat_100g
fat_100g

Conclusion

- On peut a minima définir 3 profils de base : un qui recherche des produit ayant un bon Nutri-Score, un recherchant des produits avec pas ou peu d'additifs, et un qui évite l'huile de palme.
- On peut aussi recommander des catégories de produit.

Améliorations

- Recalculer les valeurs d'énergie manquantes à partir des quantités de glucides, protéines...
- Travailler le découpage en catégories