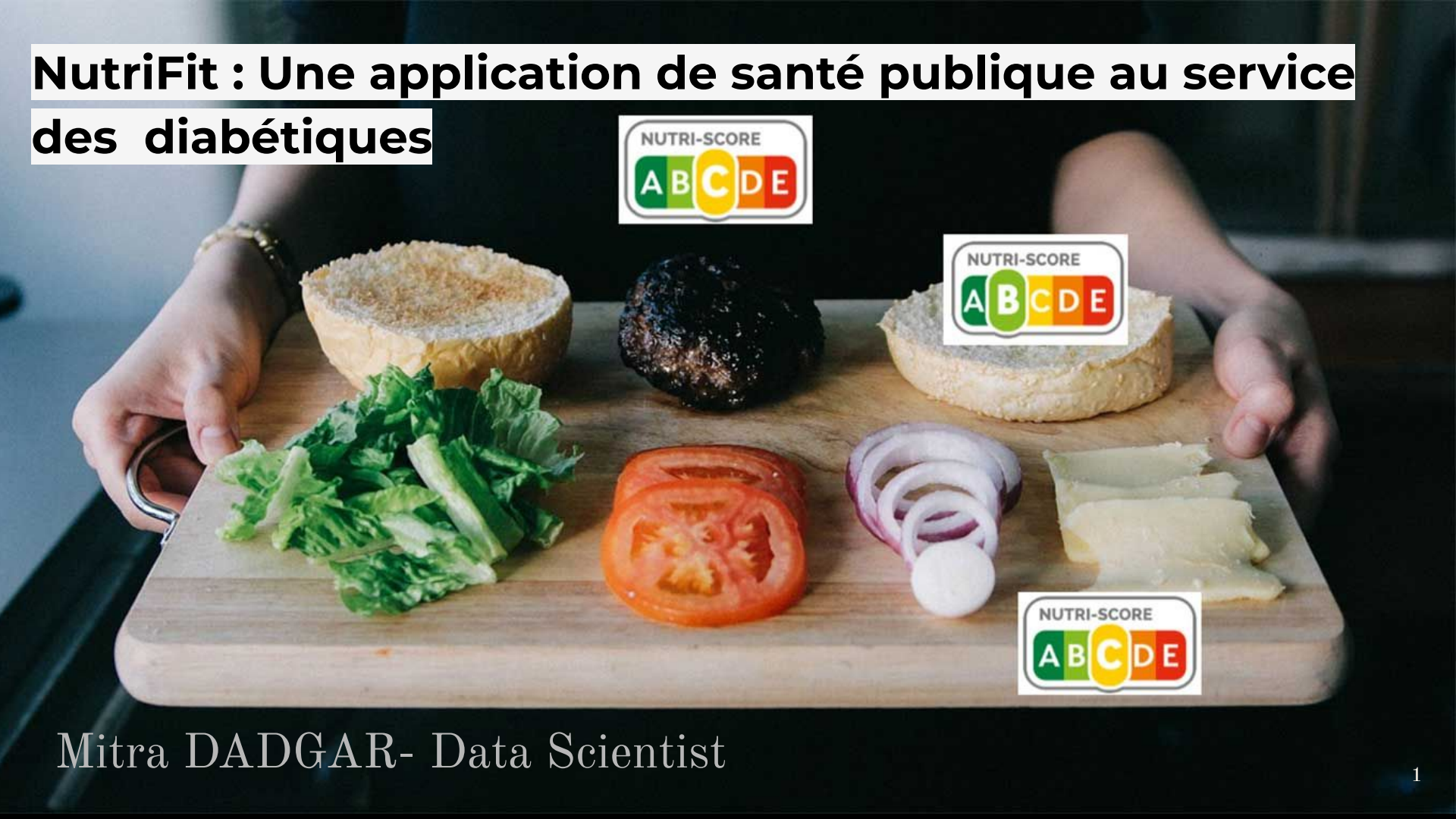
NutriFit : Une application de santé publique au service des diabétiques



Mitra DADGAR- Data Scientist

1. Présentation de l’application Nutri’Fit
2. Nettoyons les données
3. Analyses descriptives
4. Analyses multivariées
5. Construction de l’algorithme de NutriFit
6. Faisabilité et conclusion

1. L’application NutriFit

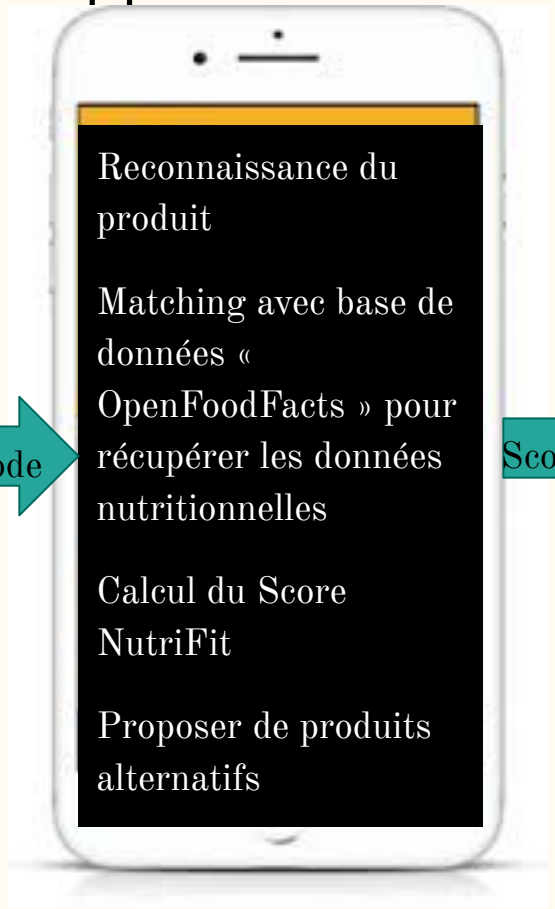
L'agence "Santé publique France" a lancé un appel à projets pour trouver des idées innovantes d’applications en lien avec l'alimentation.

Santé publique France

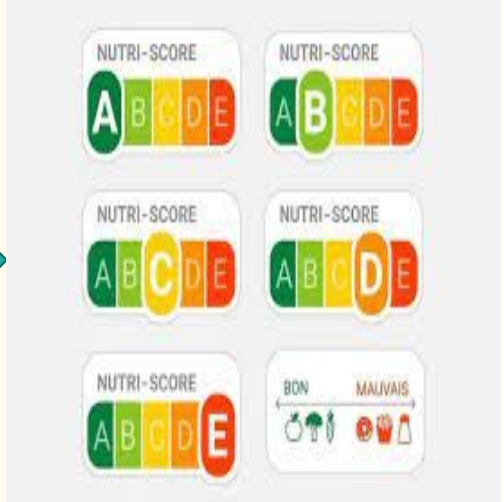
- Utilisation du code-barre pour identifier un produit
- Attribution d’un groupe sur la base d’un algorithme
- Proposer des produits alternatifs
- Projet open source donc transparence de l’algorithme

Fonctionnement de l’application

Scan Code



Score NutriFit



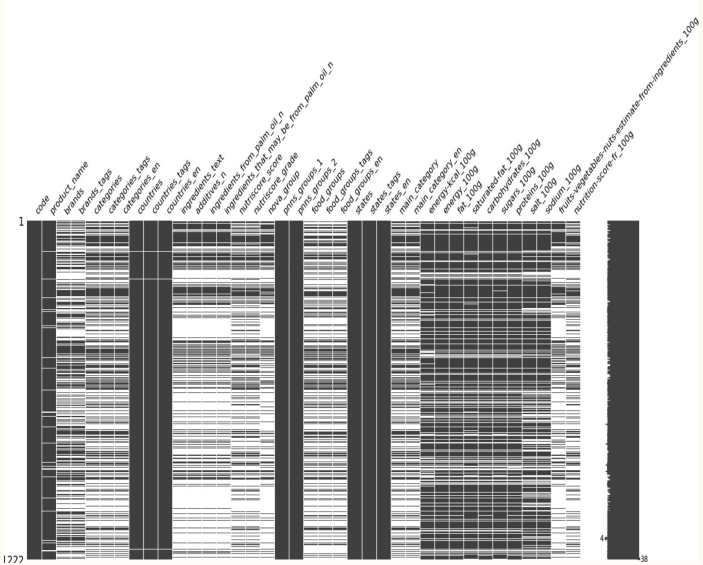
2.1 Nettoyage des données

Jeu de données assez importants (4 Gb)

```
# Taille de dataframe
df.shape
```

```
(2191222, 191)
```

- ❖ Traitement des valeurs manquantes: Choix du retrait des colonnes avec plus de 75% valeurs manquants
- ❖ Conservation de 38 colonnes



- ❖ Suppression des colonnes redondantes
- ❖ Suppression des duplicatas

```
# Détecter duplivatas
df.duplicated('code').sum()
```

```
242
```

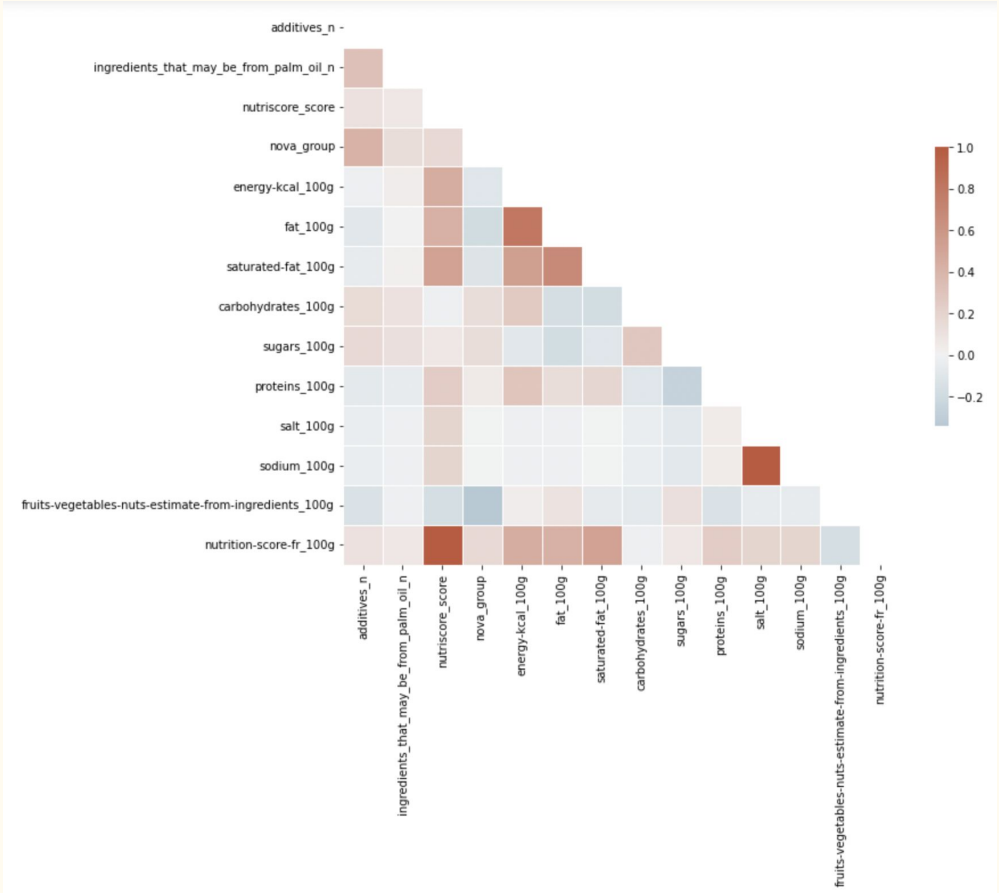
- ❖ Modification des types
- ❖ Traitement des anomalies et outliers
- ❖ Traitement des valeurs manquantes
- ❖ (fillna, IterativeImputer)

2.2 Nettoyage des données

On voit qu'il y a une forte corrélation entre "salt" et "sodium".

Il existe aussi une forte corrélation entre "nutrition-score-fr_100g" et "nutriscore_score".

On garde 'salt_100g' et 'nutrition-score-fr_100g' qui ont un mieux taux remplissage.

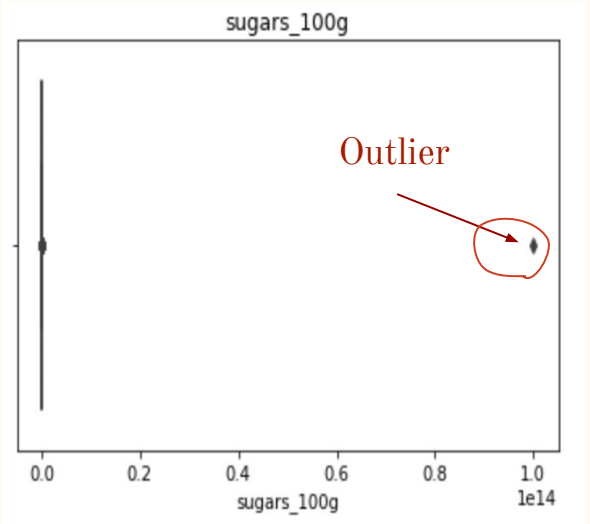


2.3 Nettoyage des données: Détection des outliers et des anomalies

- Suppression de valeurs pour l'énergie (inférieure à 0 et supérieures à 900) kcal

- Supprimer les valeurs inférieure à 0 et supérieures 100 g aux valeurs 100g

('fat_100g', 'saturated-fat_100g', 'carbohydrates_100g', 'sugars_100g', 'proteins_100g', 'salt_100g', 'sodium_100g', 'fruits-vegetables-nuts-estimate-from-ingredients_100g', 'nutrition-score-fr_100g')



- Supprimer les lignes qui ont NAN pour

('energy-kcal_100g', 'fat_100g','saturated-fat_100g', 'carbohydrates_100g', 'sugars_100g', 'proteins_100g', 'salt_100g')

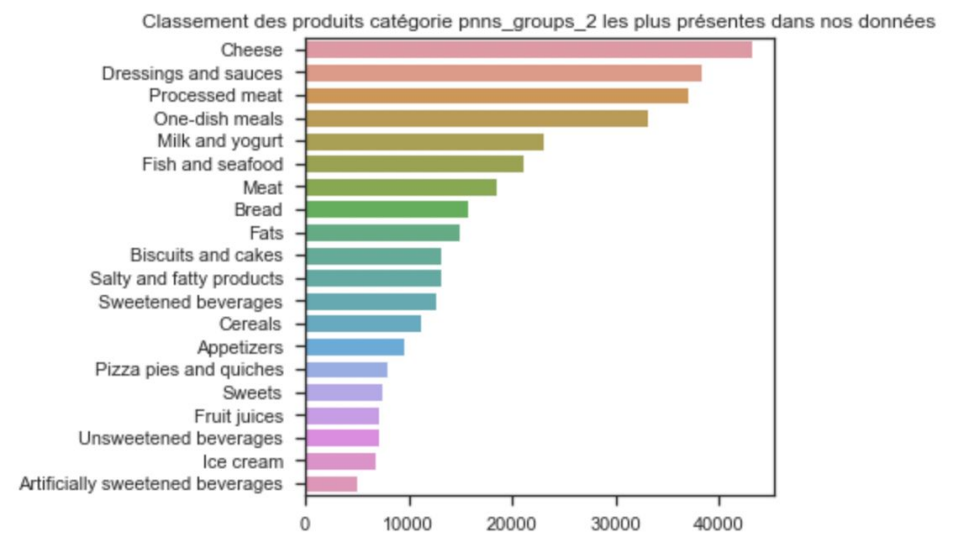
- La sommes des ingrédients ne doit pas dépasser 100 g
- Supprimer les lignes sans 'product name' et 'Countries'
- Remplacement de NaN par 'unknown'
- Utiliser IterativeImputer pour les variables numériques

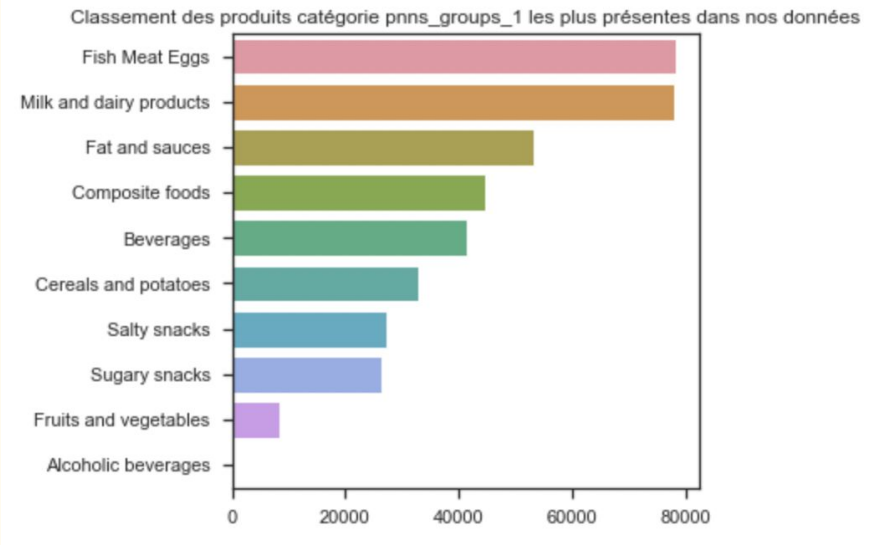
Taille de df après nettoyage
Lignes = 972636 supprimées
Colonnes = 163 supprimées

```
: df_clean.shape
(1218586, 23)
```

3. Analyse descriptive

Les produits(pnns_groups_1 et pnns_groups_2)

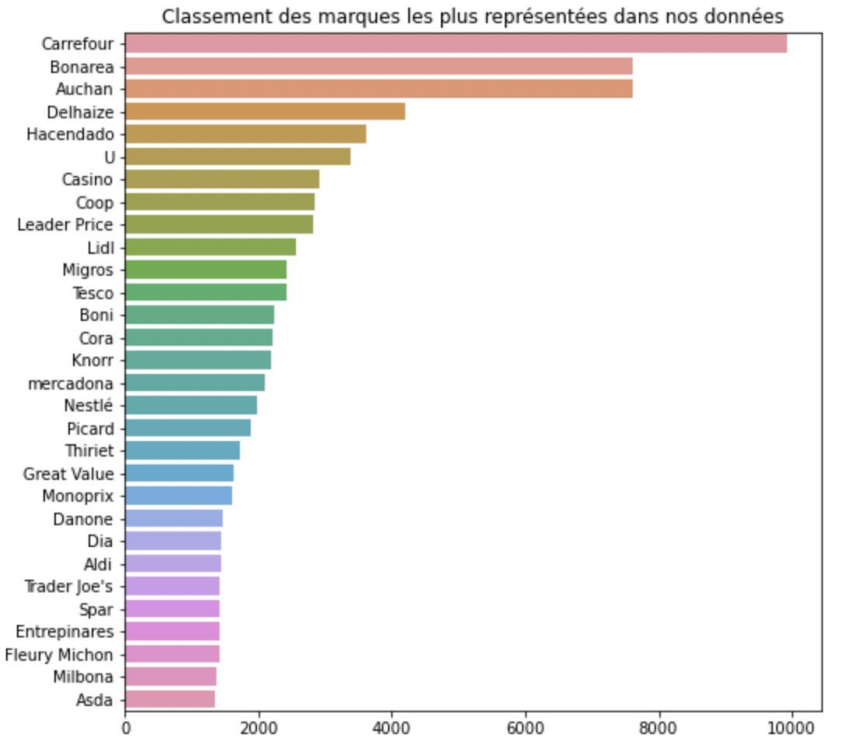




Les fromages, desserts, sauces, viandes (meat,fish) et le lait sont les produits les plus consommés.

Nombre de produits : 1218586

Les marques

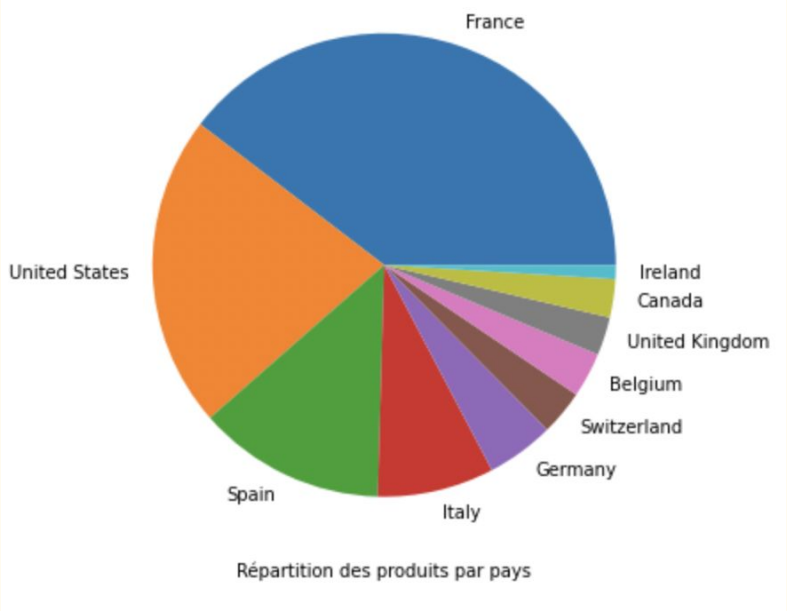


Les marques distributeurs sont les plus représentées :Carrefour, Bonarea, Auchan.

Nombre de marques : 109050

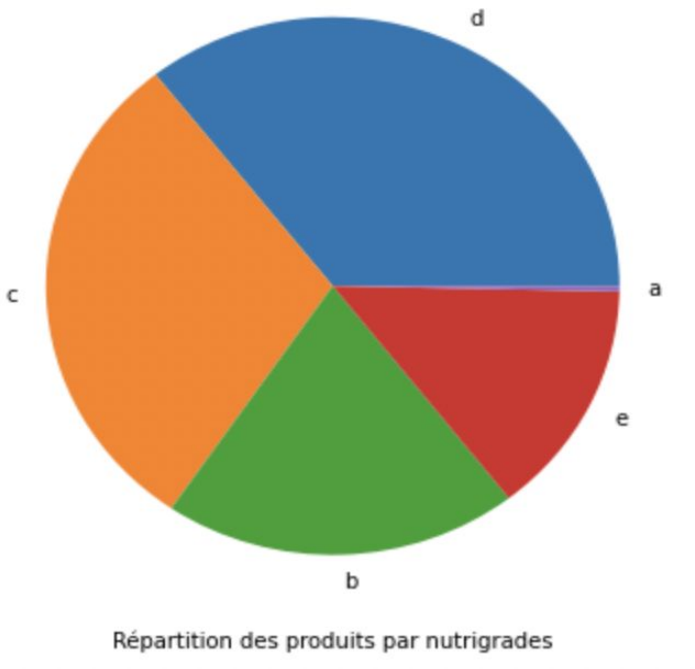
Les pays

La France est le pays où l'on retrouve le plus de produits après les États-Unis et l'Espagne



La nutrigrade

La plupart des produits sont dans la catégorie de Nutrigrade D et C et B



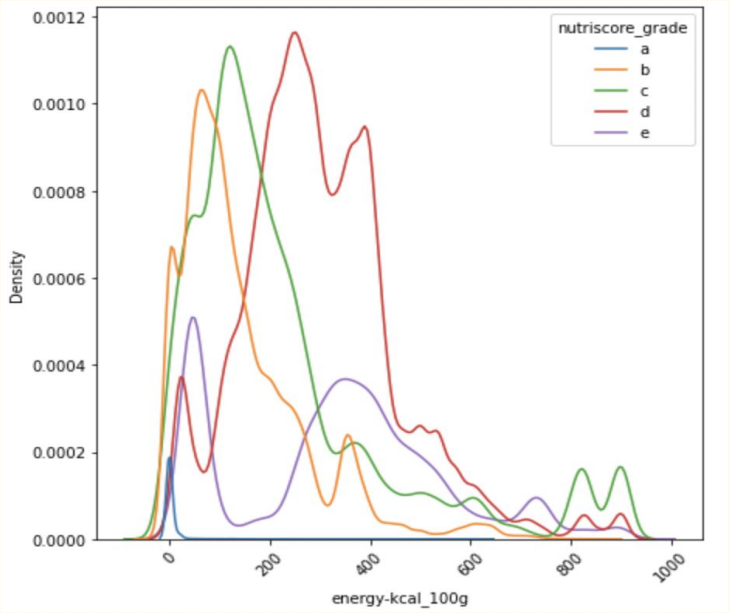
4. Analyse multivariée

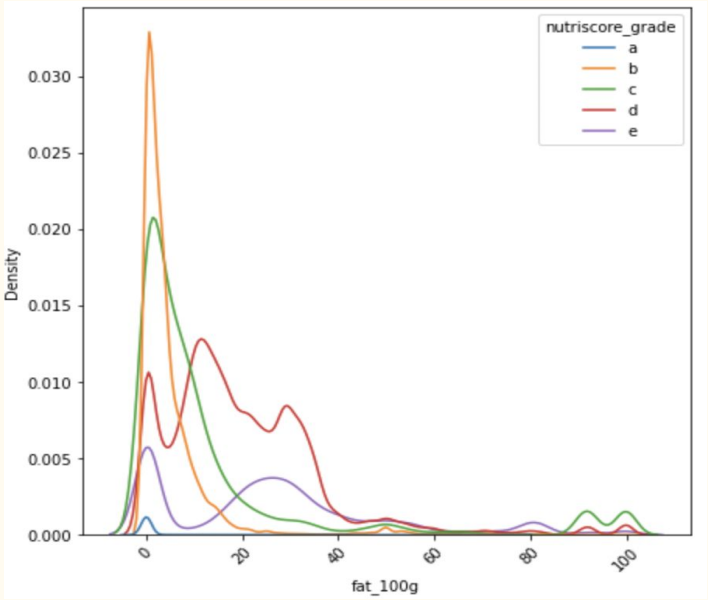
Hypothèses de travail:

- Lien entre ingrédients (sucre/fat/salt) et données nutritionnelles(nutrigrade) et énergétique

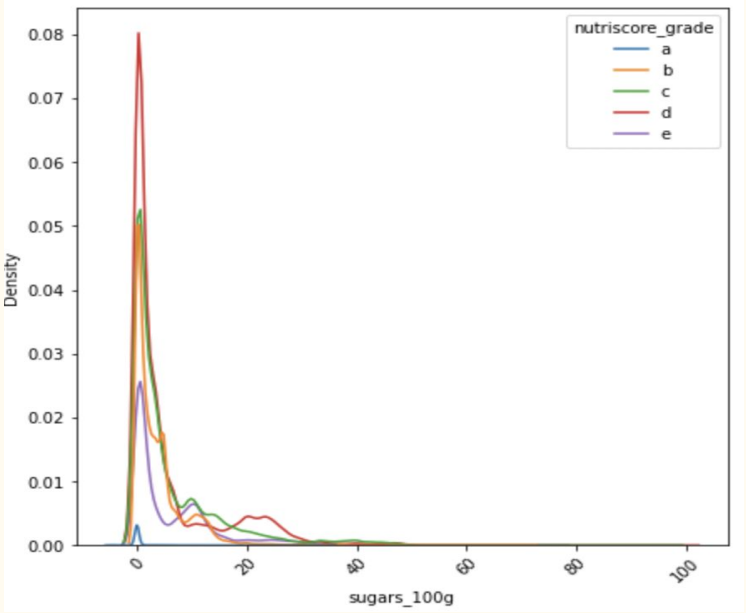
- Utilisation des données nutritionnelles, énergétiques et de la composition pour créer un indice universel

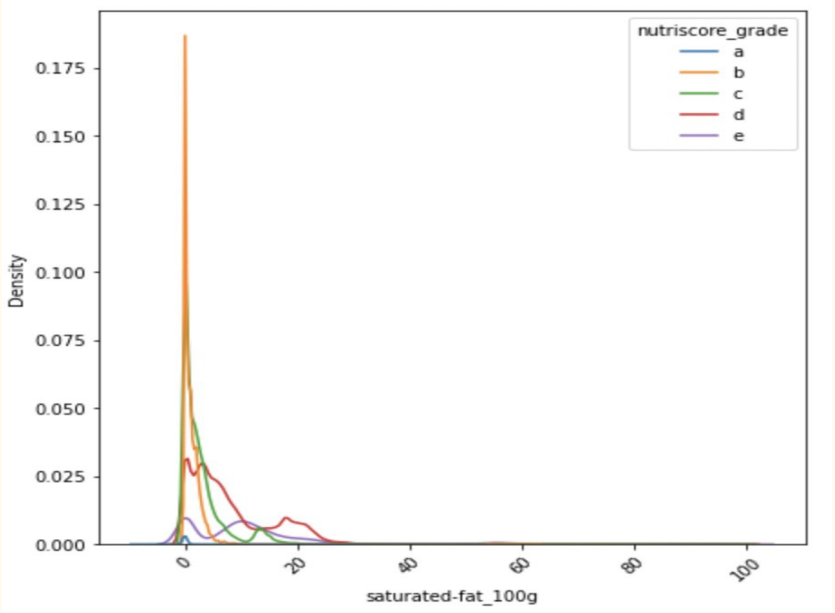
4.1 Analyse multivariée



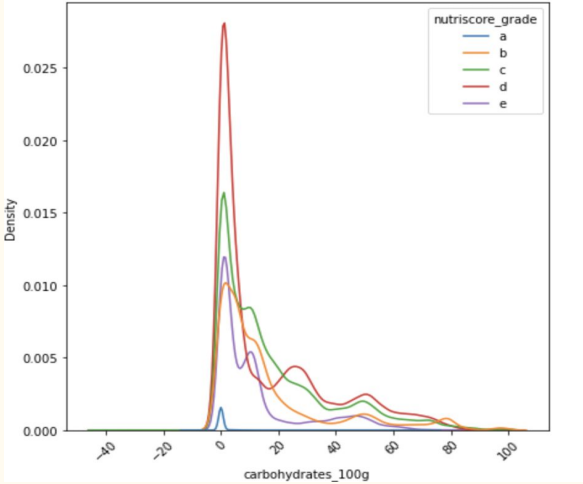


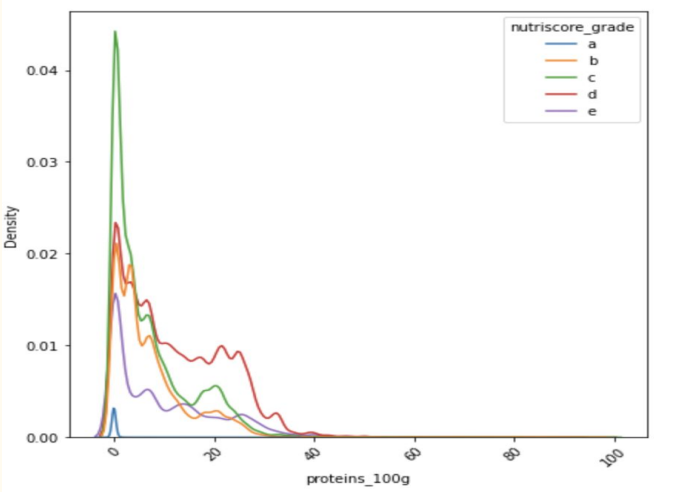
4.2 Analyse multivariée

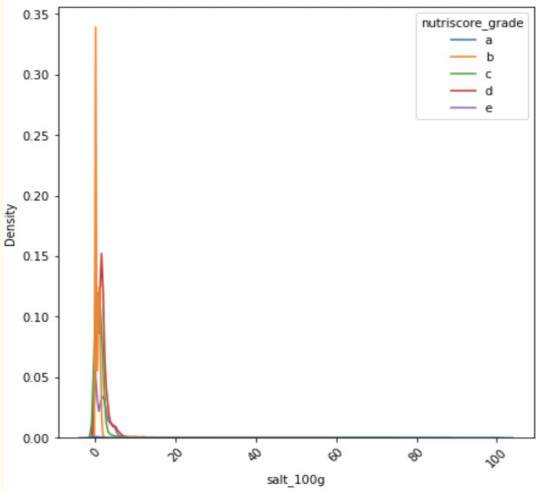


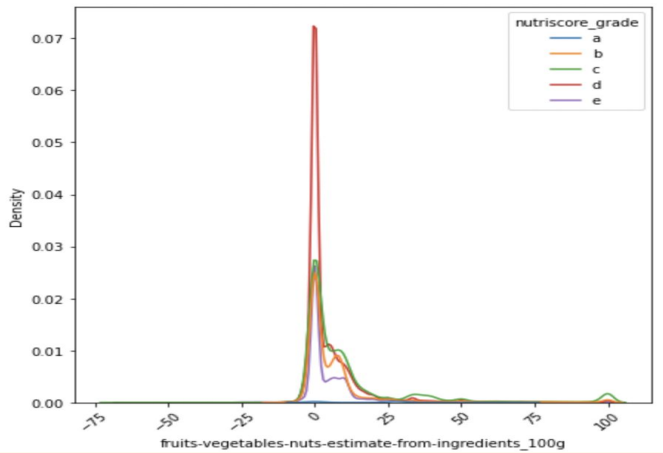


4.3 Analyses multivariées



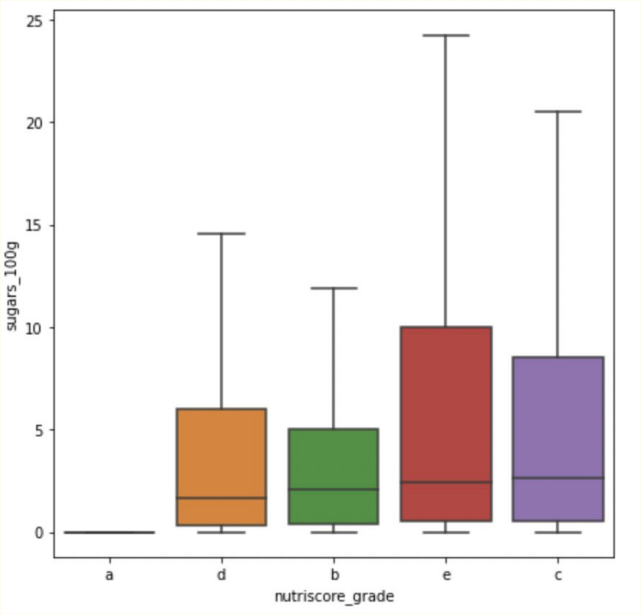


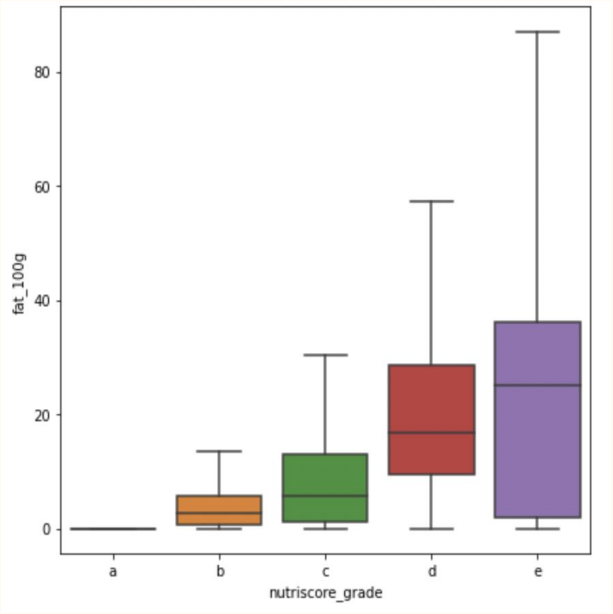


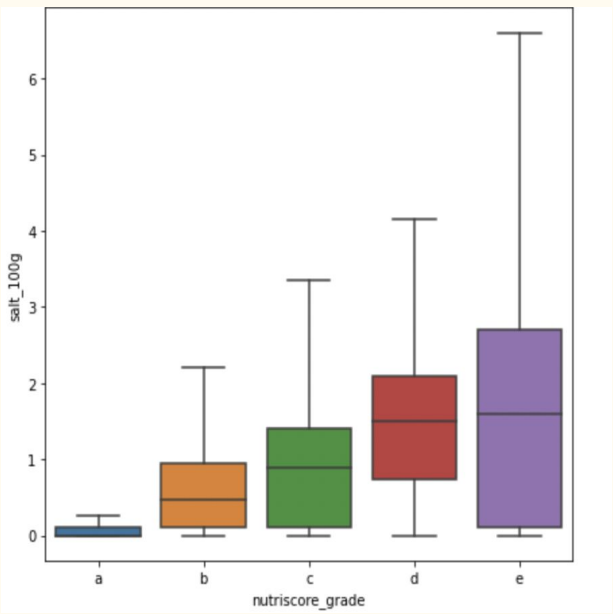


4.5 Analyse multivariée

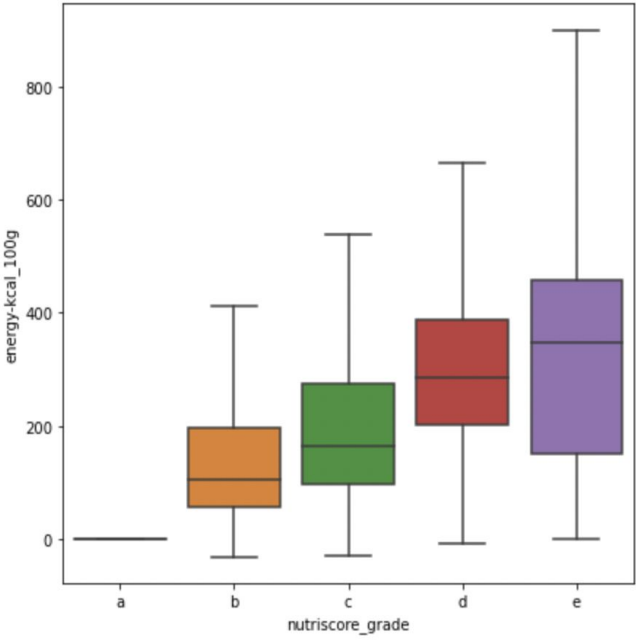
On va montrer l'influence des variables cibles (fat , sel, carboh, additif, sugar) sur le nutrigrade

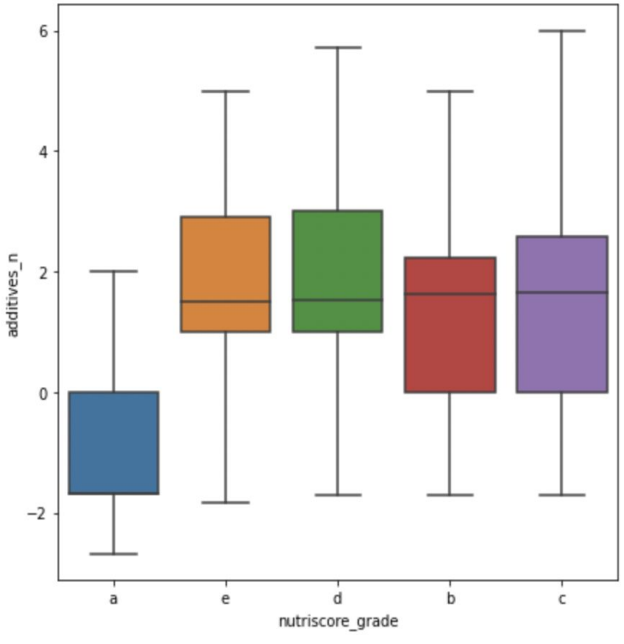


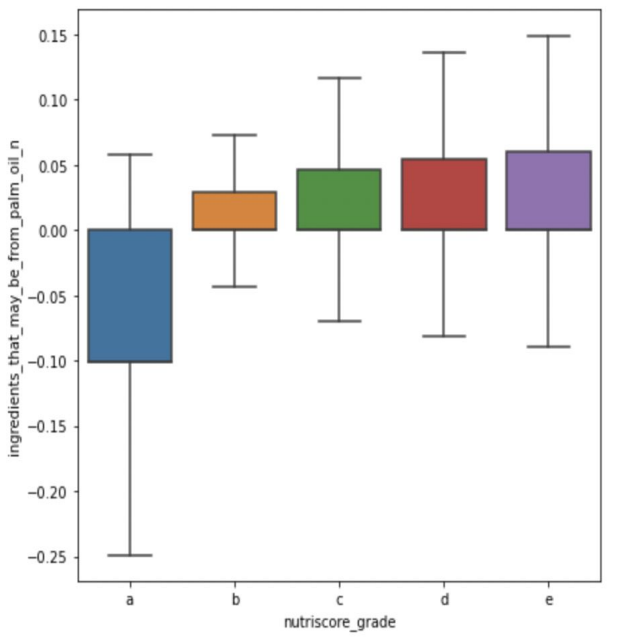




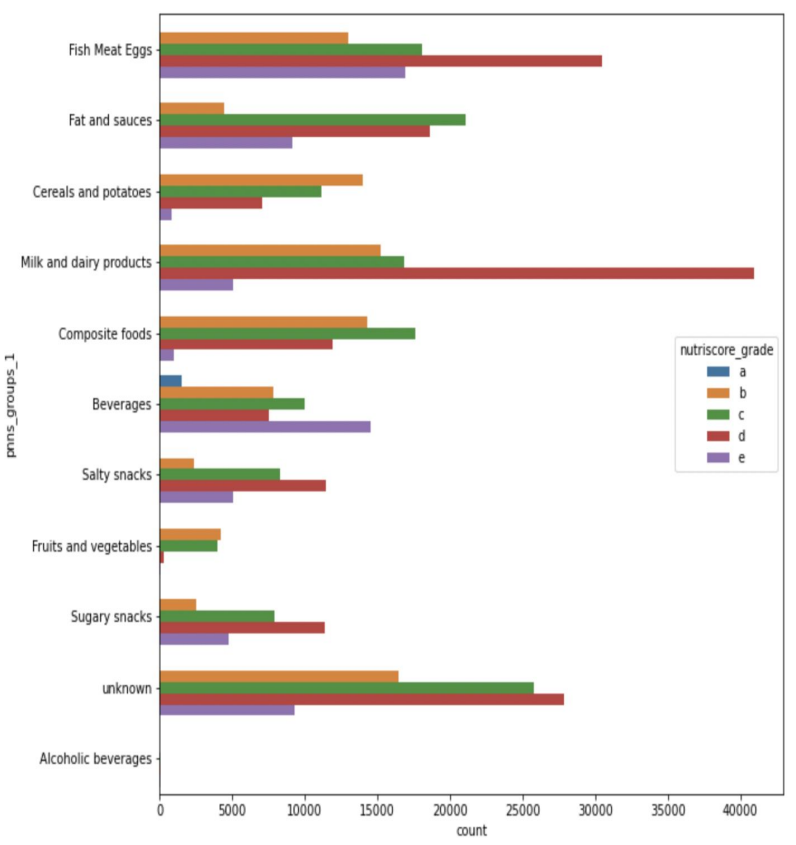
4.6 Analyse multivariée

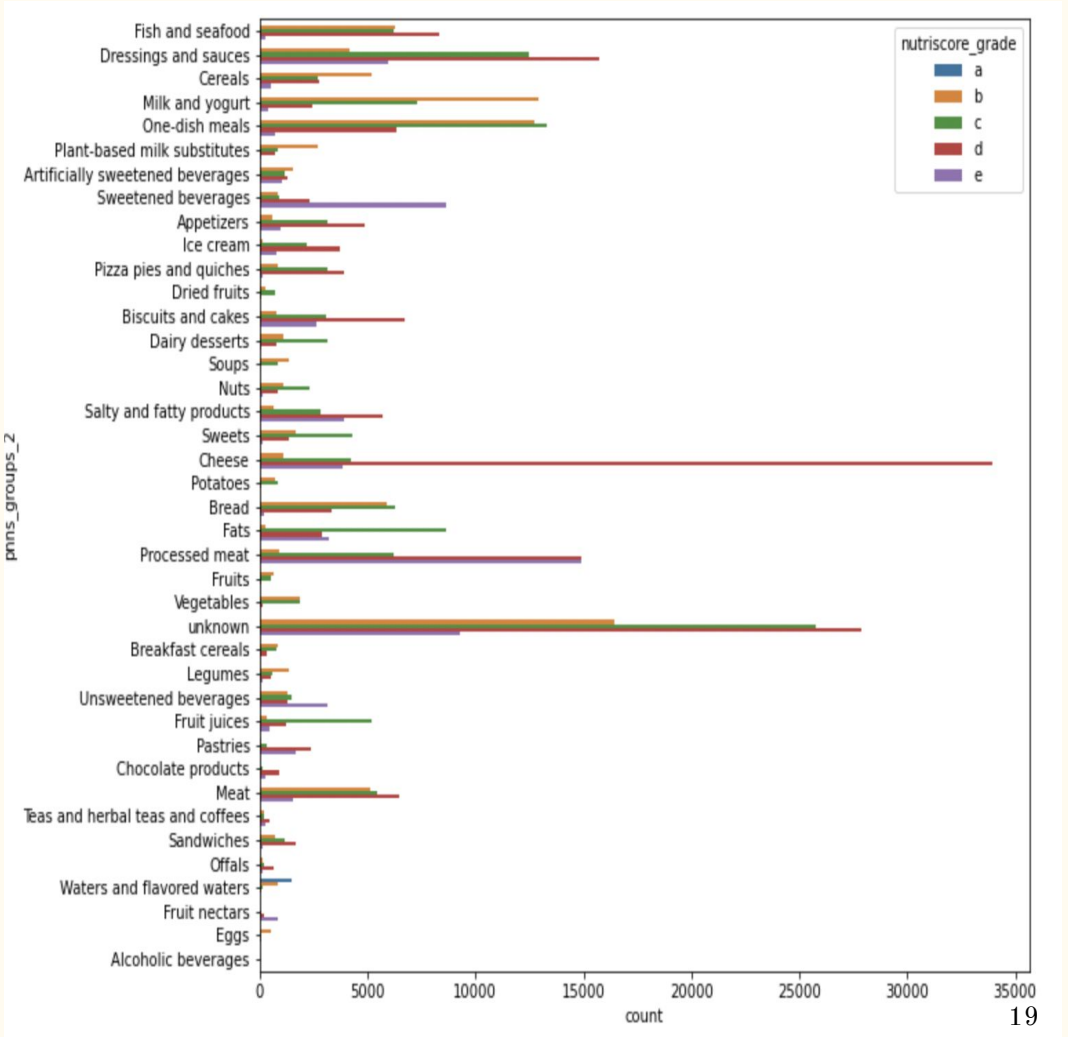






4.7 Analyses multivariées

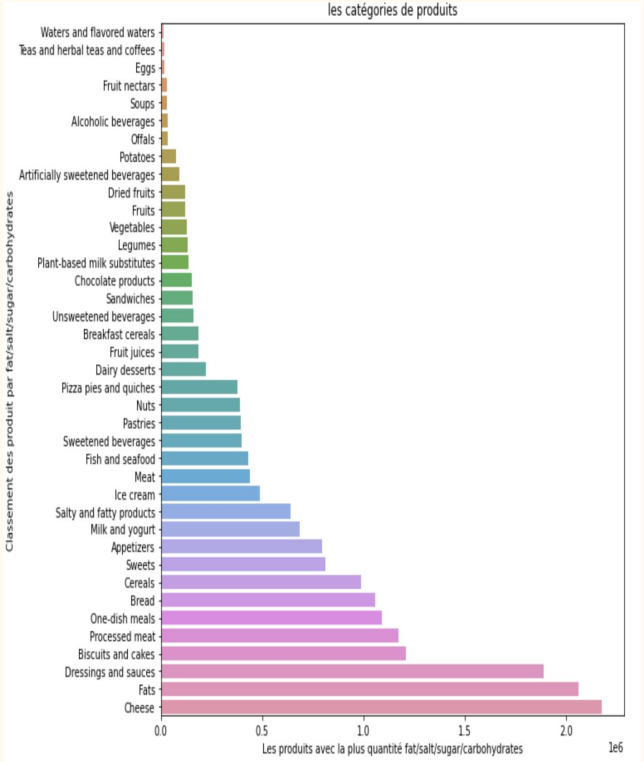


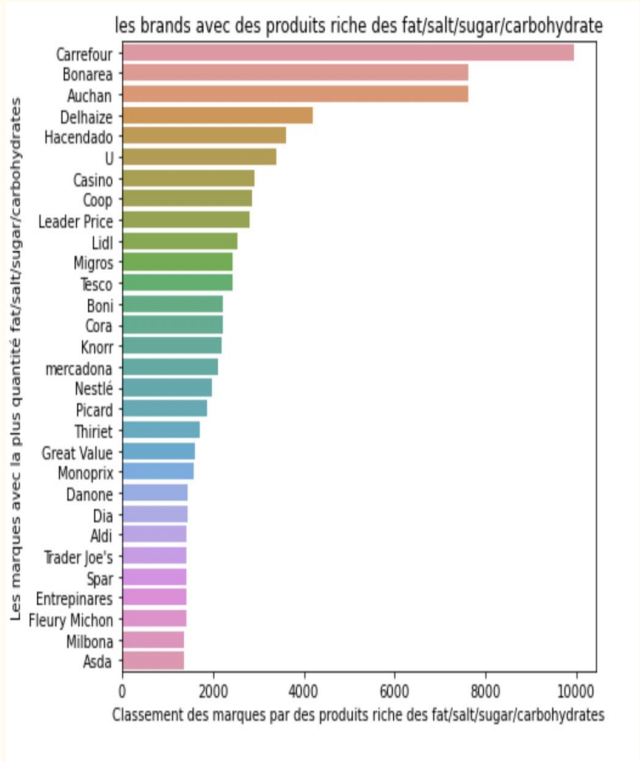


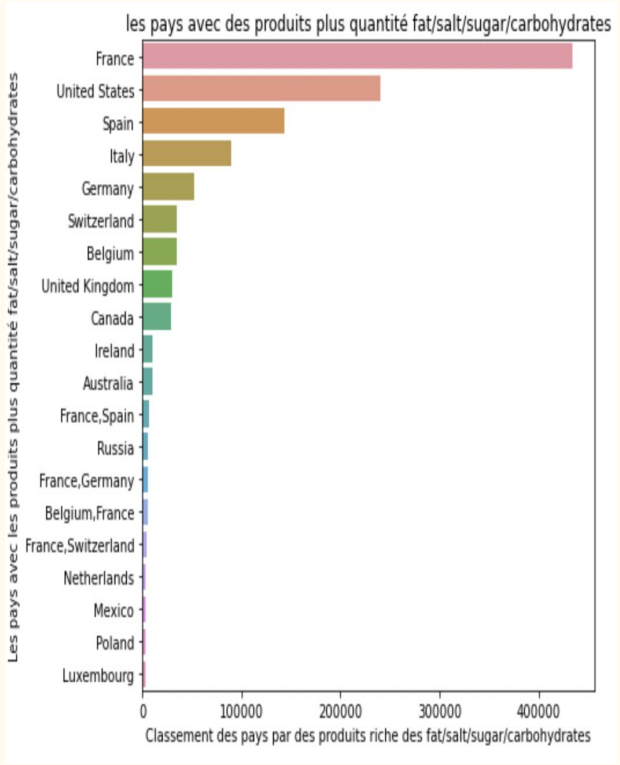
Top meilleurs catégories et marques et pays par rapport aux variables cibles

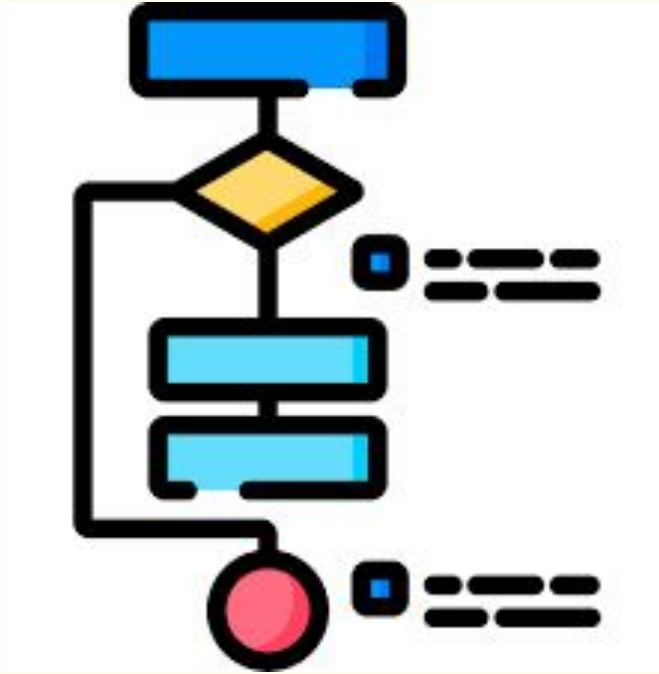
(fat_100g, saturated-fat_100g, carbohydrates_100g, sugars_100g, salt_100g)

Tri à l'aide d'une nouvelle variable qui est la somme des variables cibles



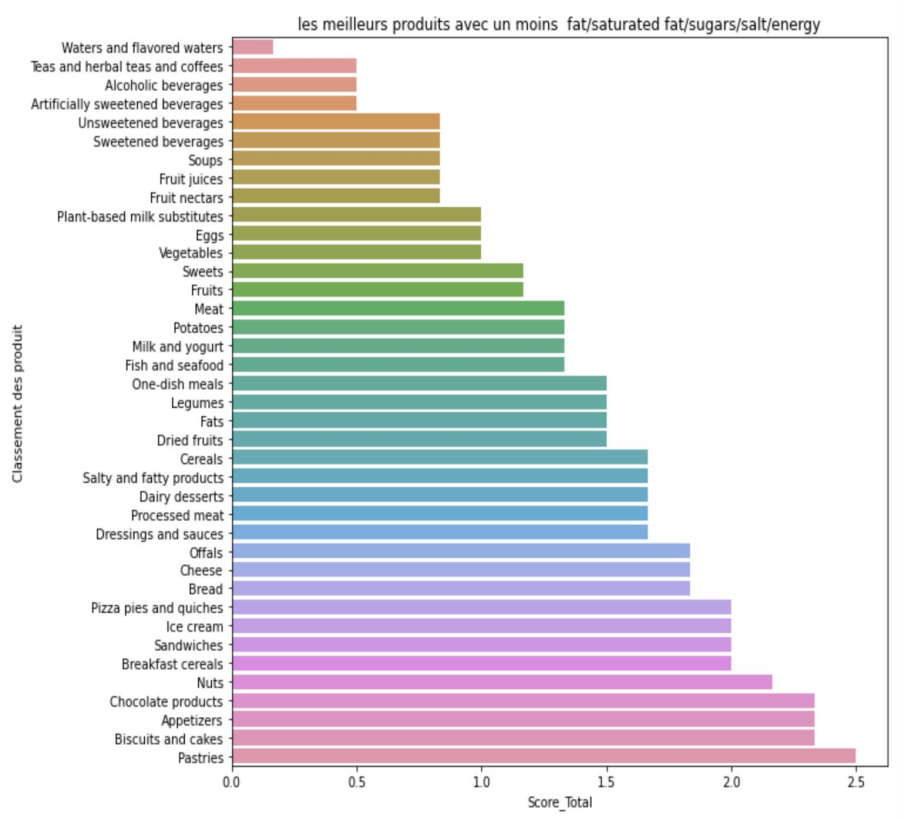






6. L’algorithme de NutriFit

Création un score général



Création d'un score (0, 3) pour chacune des valeurs nutritionnelles (discrétisation par quartile)

NutriFit : moyenne des scores

3 pour le produit avec plus fat/saturated fat/sugars/salt/energy , 0 pour le moins fat/saturated fat/sugars/salt/energy

```
df['Score_energy-kcal_100g'] = pd.qcut(df['energy-kcal_100g'], 4, labels = False,duplicates='drop')
df['Score_fat_100g'] = pd.qcut(df['fat_100g'], 4, labels = False,duplicates='drop')
df['Score_saturated-fat_100g'] = pd.qcut(df['saturated-fat_100g'], 4, labels = False,duplicates='drop')
df['Score_carbohydrates_100g'] = pd.qcut(df['carbohydrates_100g'], 4, labels = False,duplicates='drop')
df['Score_sugars_100g'] = pd.qcut(df['sugars_100g'], 4, labels = False,duplicates='drop')
df['Score_salt_100g'] = pd.qcut(df['salt_100g'], 4, labels = False,duplicates='drop')
df["Score_Total"] = (df["Score_energy-kcal_100g"] + df['Score_fat_100g'] + df['Score_saturated-fat_100g'] + df['Scor
```

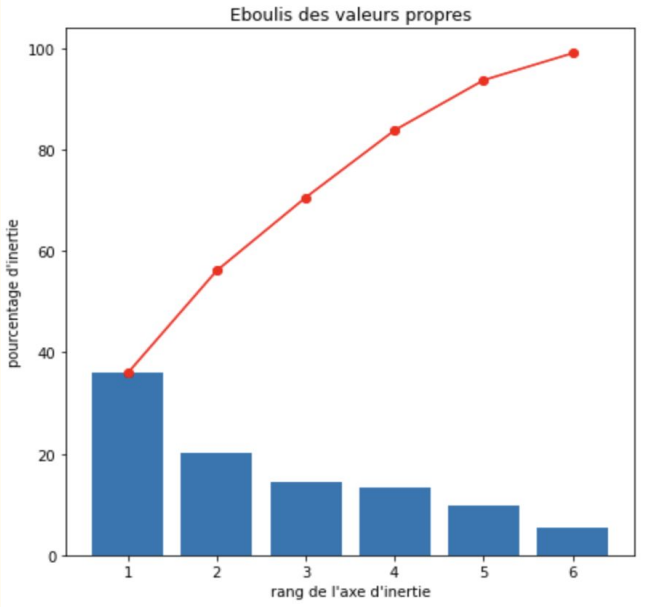
ACP et réduction de dimensions

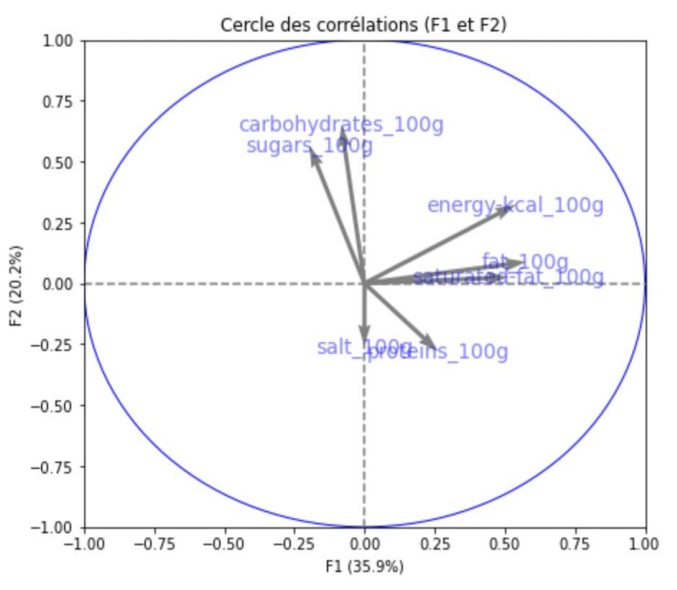
nous souhaitons trouver une nouvelle façon de synthétiser les caractéristiques communes à la majorité des données et trouver la combinaison optimale des composantes principales

On extraire les variables 'energy-kcal_100g', 'fat_100g','saturated-fat_100g', 'carbohydrates_100g', 'sugars_100g', 'salt_100g', 'proteins_100g' Puis standardiser ces variables

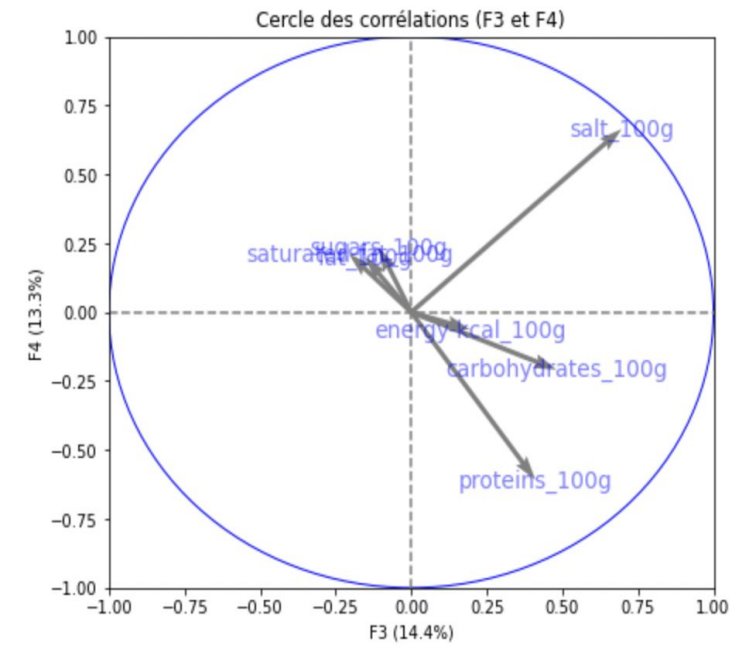
F1 (35.9%) : Energie en calories et lipides et graisses saturées(fat et saturated_fat)

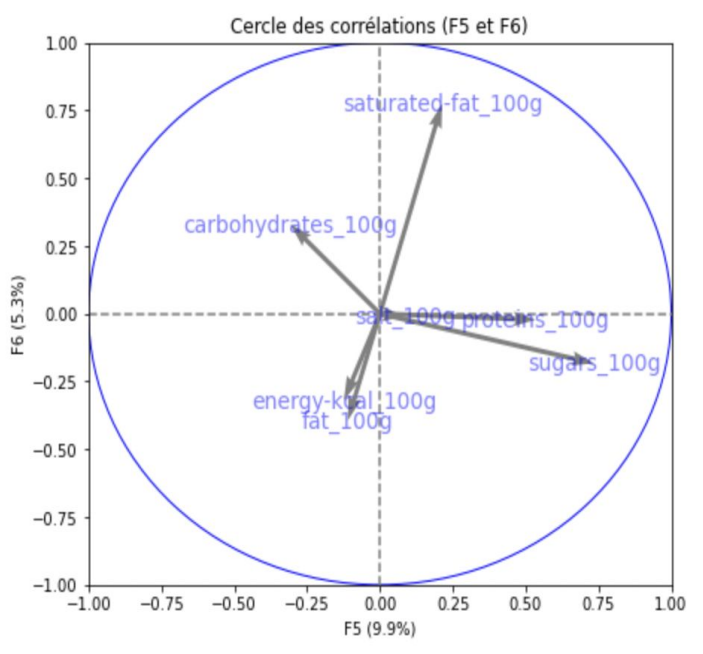
F2 (20.2%) : Sel, Protéines, Fibres, Carbohydrates, Sucre => 56.1% de variance expliquée



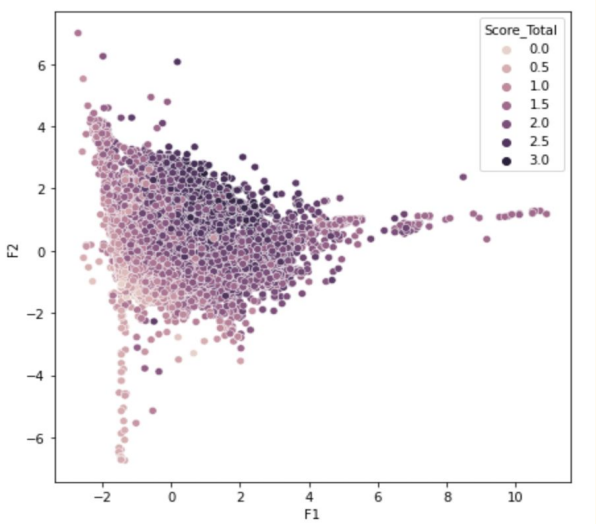


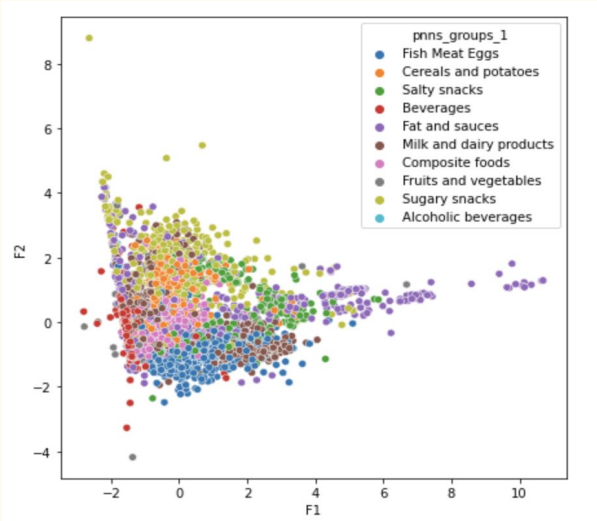
Le cercle des corrélations des composantes F3,F4 et F5,F6

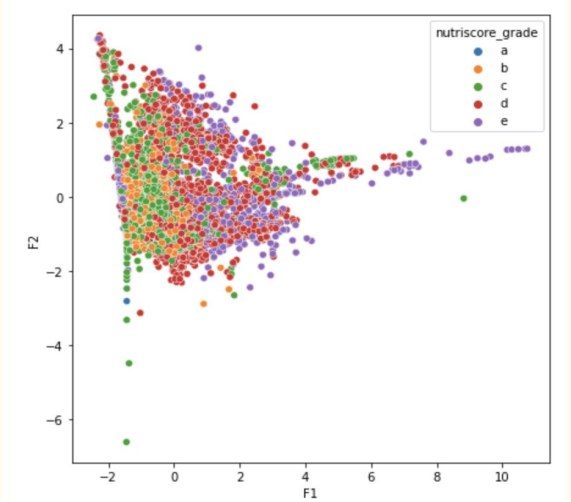


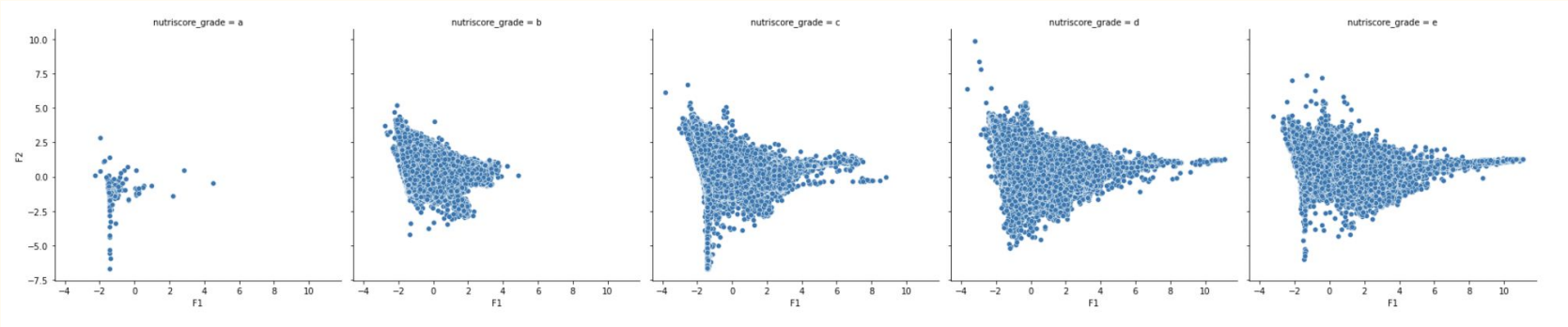


Projeter les points sur les composantes F1 et F2, puis isoler les valeurs par score_total et colorez par nutriscore_grade et catégories des produits

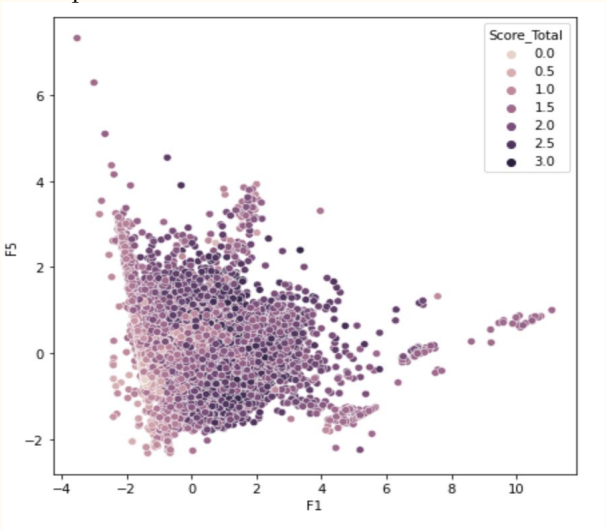


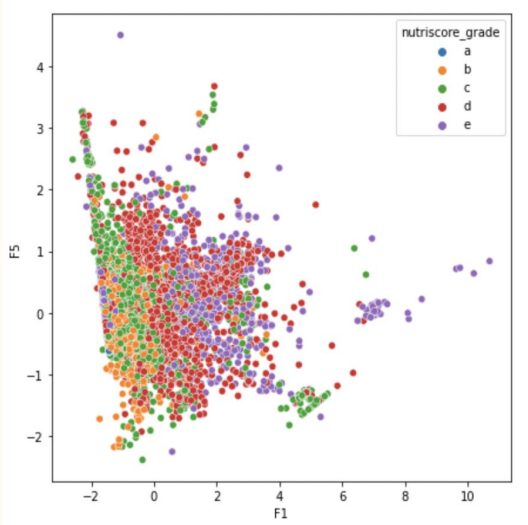


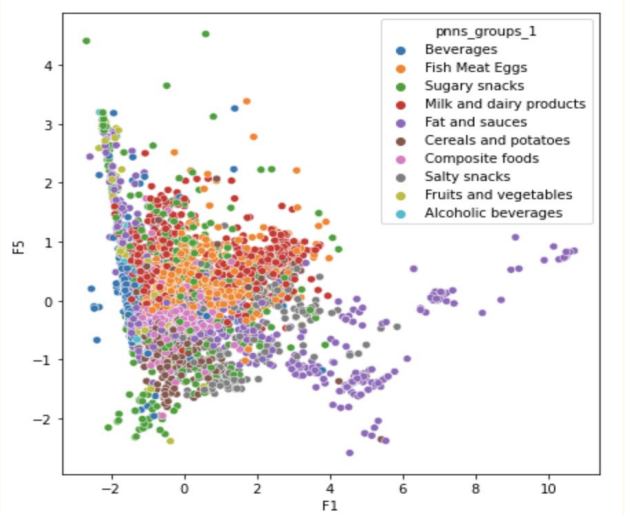


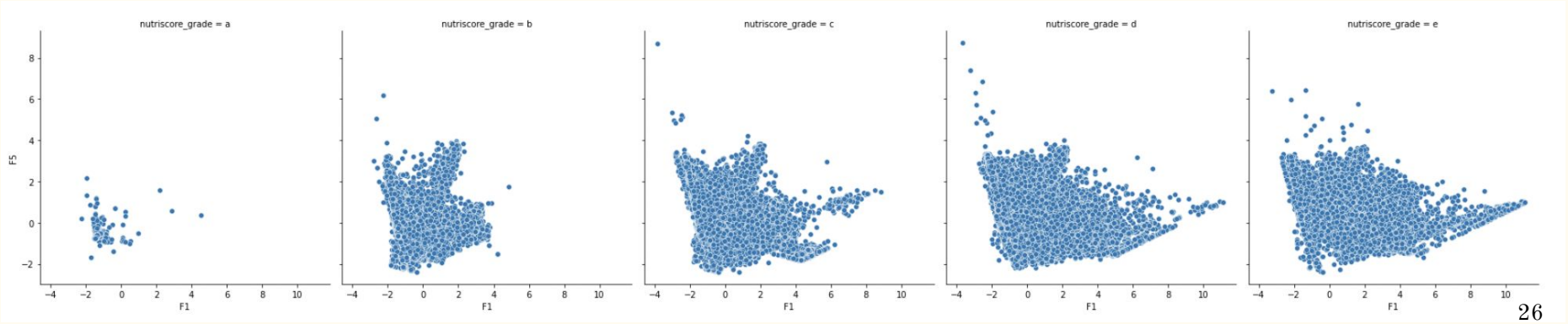


Projeter les points sur les composantes F1 et F5, puis isoler les valeurs par score_total et colorez par nutriscore_grade et catégories des produits









Test de Kruskal Wallis

Utilisé pour déterminer s'il existe ou non une différence statistiquement significative entre les médianes de cinq groupes (pour la variable score total)

En fonction de la note on a des groupes a,b,c,d,e; si la composition de Score-Total(le moyen des score par quartiles pour fat/saturated fat/sugars/salt/energy) justifier que les observatoires sont dans les groupe différentes.

Question: Est ce que les produits sont issus d'une même population sur la base du score ou pas ?

- L'hypothèse nulle (H0) : la médiane de score_total est similaire dans tous les groupes.
- L'hypothèse alternative : (Ha) : La médiane de score_total n'est pas similaire dans au moins deux groupes.

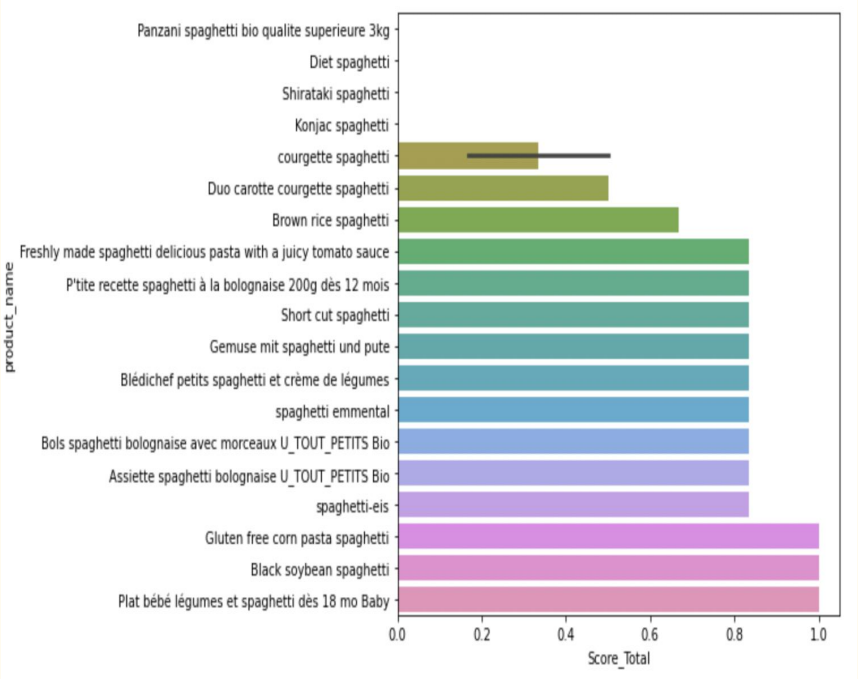
Dans ce cas, la p-value correspondante est de 0,0. Puisque cette p-value est inférieure à 0,05 (pour 95% de confiance), nous pouvons rejeter l'hypothèse nulle H0.

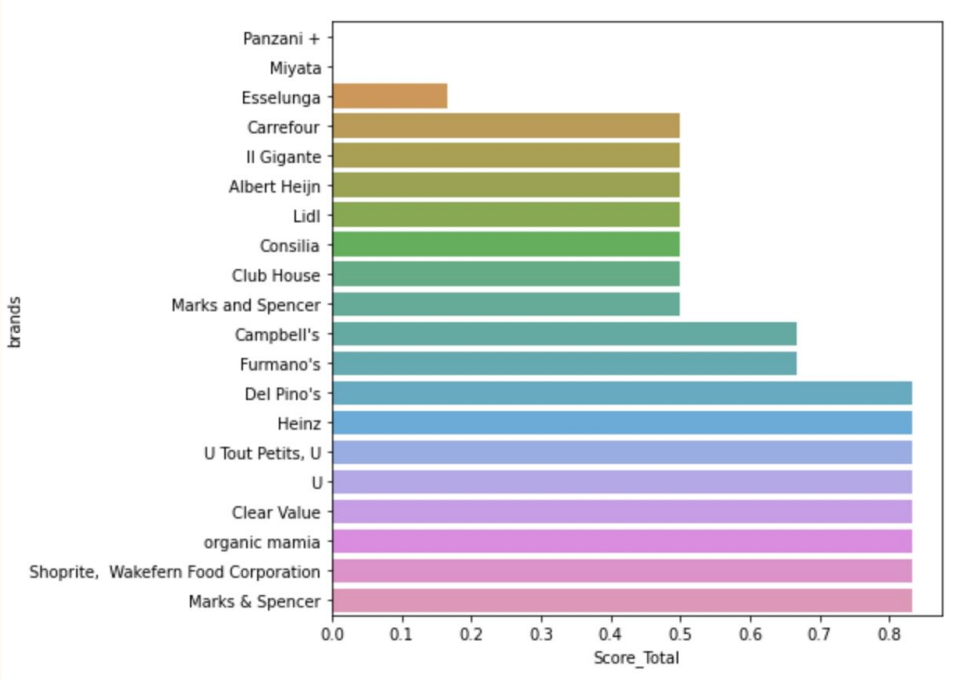
```
]: from scipy import stats
   list_var = []
   for note in ['a', 'b', 'c', 'd', 'e']:
       filtre_grade = df['nutriscore_grade'] == note
       list_var.append(df[filtre_grade]['Score_Total'].tolist())
   stats.kruskal(*list_var)
```

```
KruskalResult(statistic=123290.49942947426, pvalue=0.0)
```

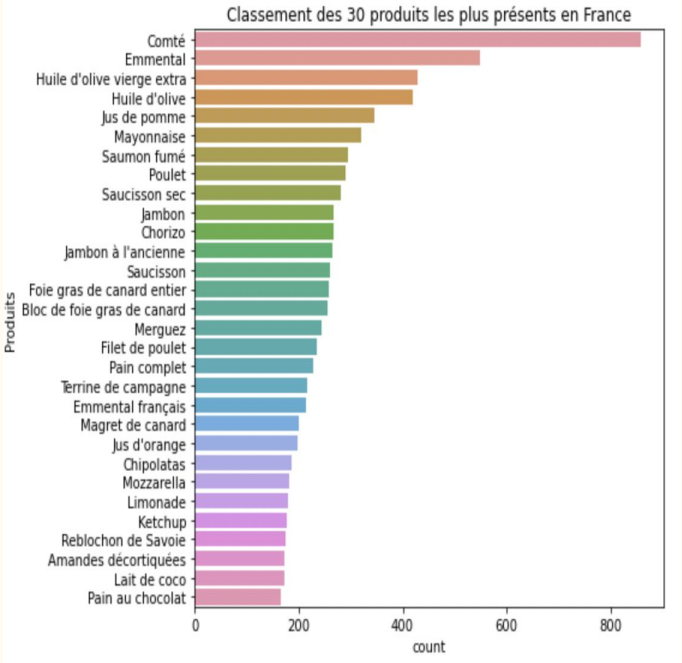
Proposer les meilleurs produits et brands selon notre recherche

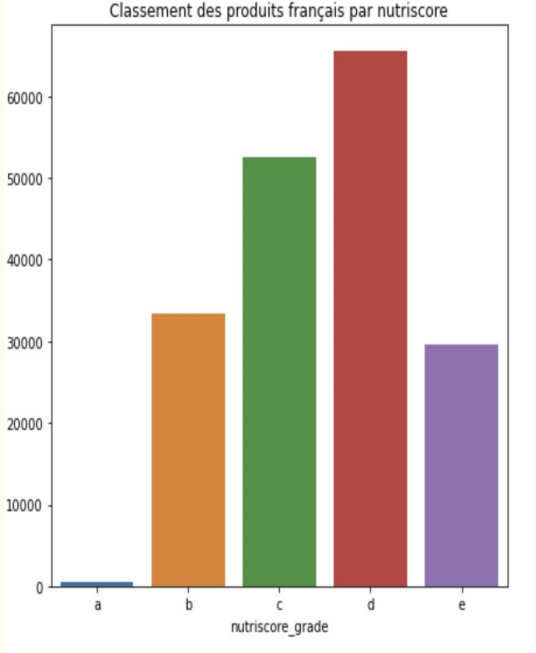
Nutri'Fit : Trouver un spaghetti avec un bon score et propose un bon brand

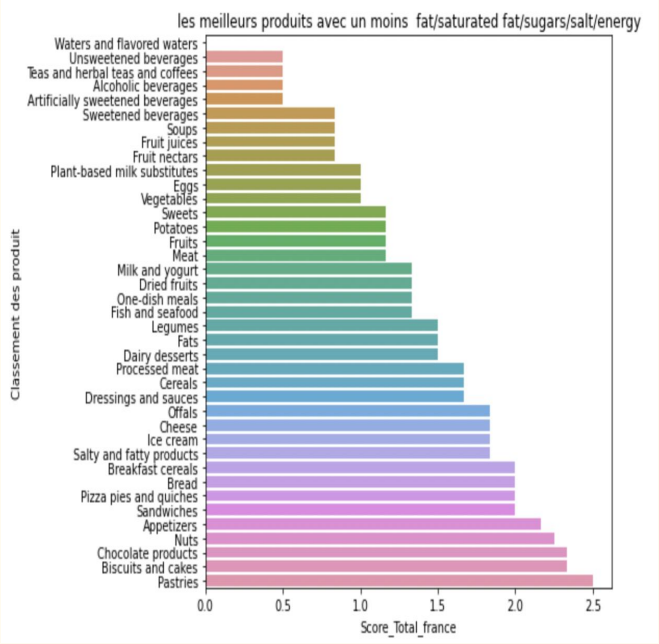




Création d'un score général pour les produits français

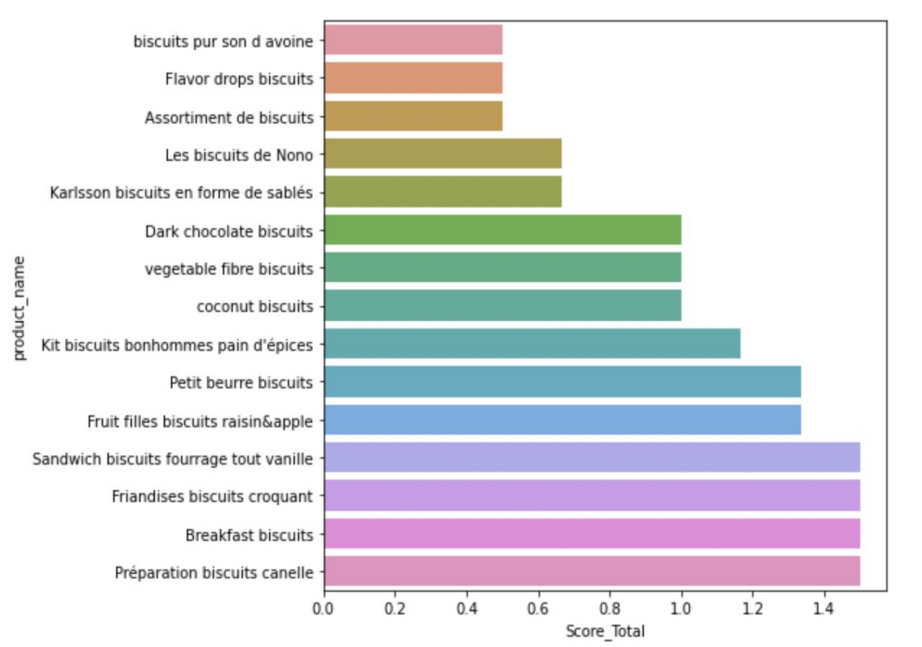


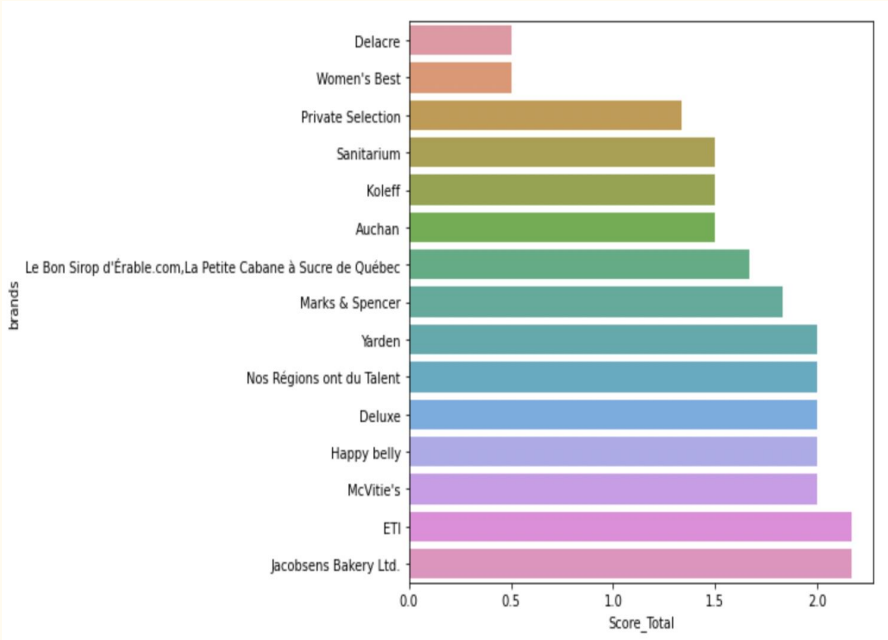




Proposer les produits français

On définit une fonction afin de nous proposer les meilleurs produits francais dans la catégorie qu'on cherche.
Par exemple: les meilleurs biscuits français avec moins de fat_100g, saturated-fat_100g, carbohydrates_100g, sugars_100g, salt_100g





6. Faisabilité et conclusion

- Projet réalisable(On a une large base de données)
- Possibilité de discriminer des produits en catégorie selon les données nutritionnelles, énergétiques et compositions
- Création d'un score entre (0-3) qui permet de savoir à quel point notre produit est bon
- Application interactif: possibilité de proposer le meilleur produit selon notre recherche entre les produits internationale ou les produits français

Pistes d’amélioration

- Intégrer la reconnaissance d’un produit par son code-barre pour pouvoir retrouver ces données
- Extraction des informations nutritionnelles à partir d’une photo