Projet 3 : Concevez une application au service de la santé publique

Date : 18/04/2023

Version 1

Sommaire

Santé publique France

1. Idée d'application

- Indicateur de nutriscore pour un utilisateur qui n'aurait que quelques informations élémentaires sur le produit (jeu de données réduit)

  Calcul automatique de nutriscore

OPEN FOOD FACTS

NUTRI-SCORE

A B C D E

2. Nettoyage effectué - fonctions

Découpage du processus de nettoyage

- Contrôle des colonnes
- Plusieurs fonctions de nettoyage particulières
- Une fonction générale appliquant toutes les fonctions de nettoyage
- Capture d'exceptions via try/except
- Sauvegarde d'un fichier nettoyé

2. Nettoyage effectué - détail

Correction des types / format des dates

```
if column[-2:] == '_t':
    new_column = column[:-2]
    dataframe[new_column] = pd.to_datetime(dataframe[column],
                                           unit='s')
    dataframe = dataframe.drop(column, axis=1)
```

Traitement des colonnes tags : mapping

```
#traces_tags
mapping = {'nuts' : 'arachides',
       'milk' : 'lait',
       'gluten' : 'gluten',
       'soybeans' : 'graines de soja',
       'peanuts' : 'arachides',
      'eggs' : 'oeufs'}
dataframe['traces_tags'] = dataframe['traces_tags'].apply(categorize, args=[mapping])
dataframe['traces_tags'] = dataframe['traces_tags'].astype('category')
print(dataframe['traces_tags'].unique())
```

2. Nettoyage effectué - détail

- **Pays d'origine**
  - France uniquement
  - Suppression nutrition-score-uk_100g
- **Suppression des informations en doublon**
- **Titres des colonnes**

```
for column in columns:
    if column[0] == '-':
        column = column[1:]
```

2. Nettoyage effectué - détail

Etude uni/multi-variée des outliers

Outliers sur 1 dimension (1% extrême)

La méthode des 1% extrêmes consiste à identifier les valeurs qui se trouvent en dehors de la plage des 1% des valeurs les plus extrêmes de l'ensemble de données. Pour cela, on peut calculer les seuils inférieur et supérieur en utilisant la formule suivante :
seuil inférieur = quantile(ensemble de données, 0.01)
seuil supérieur = quantile(ensemble de données, 0.99)

Outliers sur plusieurs dimensions (distance de Minkowski)

La méthode de Minkowski est une méthode pour mesurer la distance entre deux points dans un espace multidimensionnel. Elle utilise une formule qui prend en compte la différence de chaque dimension entre les deux points, élevée à une puissance p, et en fait la somme. Cette méthode est couramment utilisée en science des données pour le clustering et la classification.

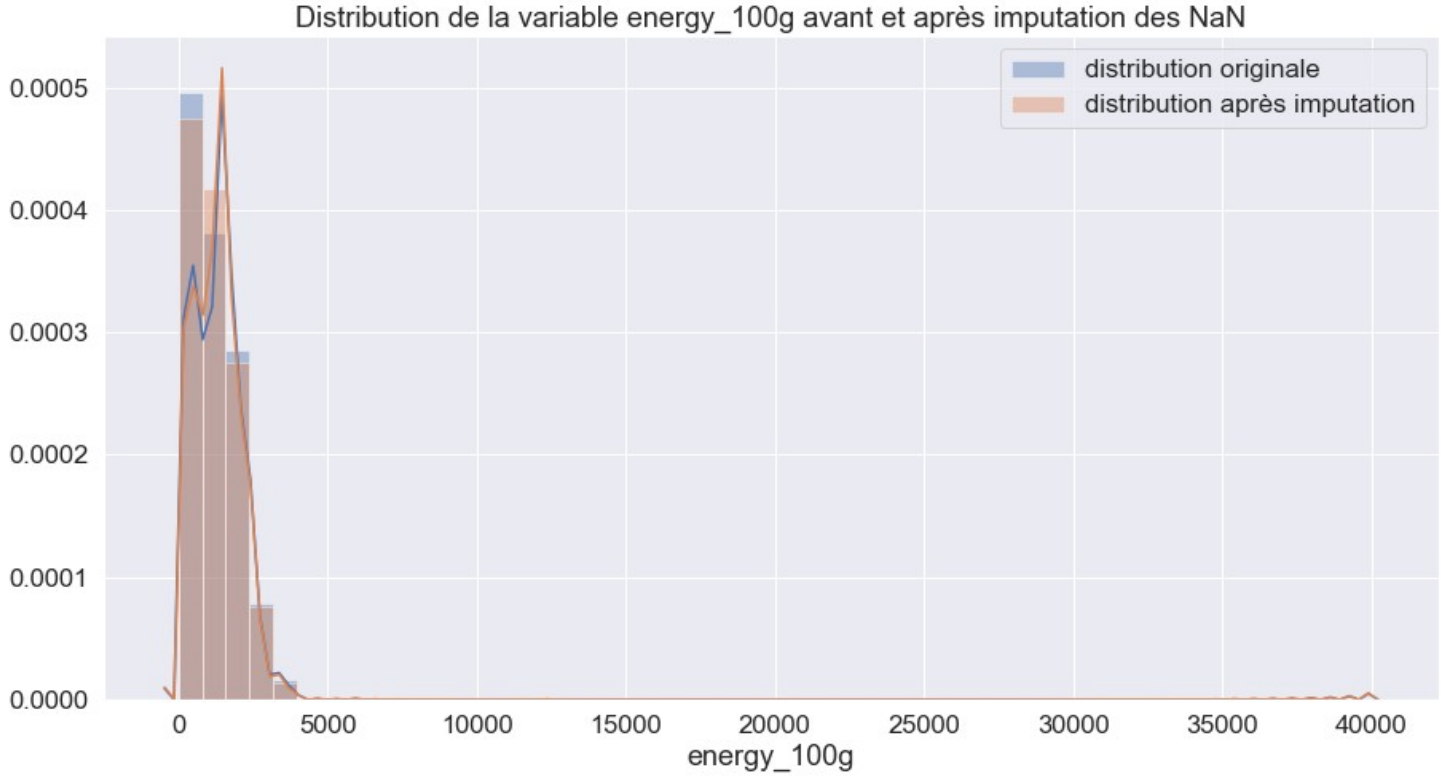
```
#outliers éloignés par rapport à leurs voisins
numeric_data = dataframe.select_dtypes(['int32', 'float64']).copy().dropna()
kdt = KDTree(numeric_data, leaf_size = 40, metric='minkowski')

dist, ind = kdt.query(numeric_data, k=3, return_distance=True)
numeric_data['3N_distance'] = np.sum(dist, axis=1)
numeric_data = numeric_data[numeric_data['3N_distance'] < numeric_data['3N_distance'].quantile(0.99)]
index_to_drop = numeric_data.index.tolist()

return dataframe.drop(index_to_drop, axis=0)
```

2. Nettoyage effectué - détail

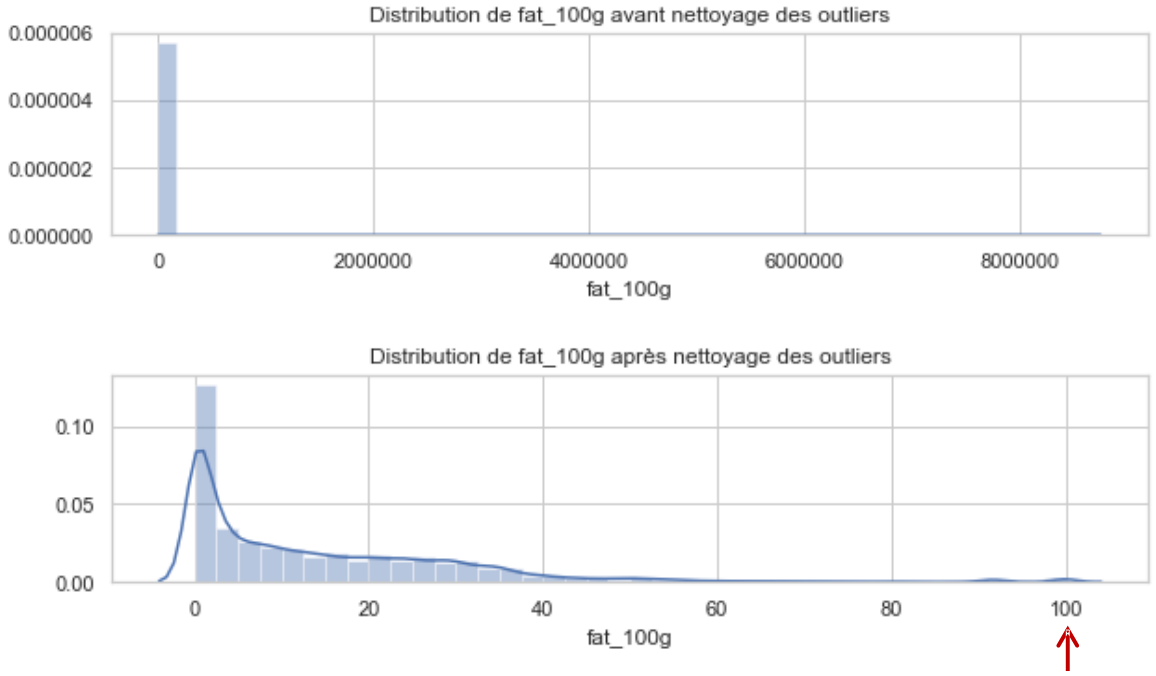
- **Traitement des NaN**
  - Suppression de colonne au delà d'un seuil préalablement fixé *(ajusté ici à 80 % maximum de taux de NaN)*
  - Imputation par la méthodes des K plus proches voisins



2. Nettoyage effectué - détail

Etude uni/multi-variée des outliers - Exemple

Purge des outliers

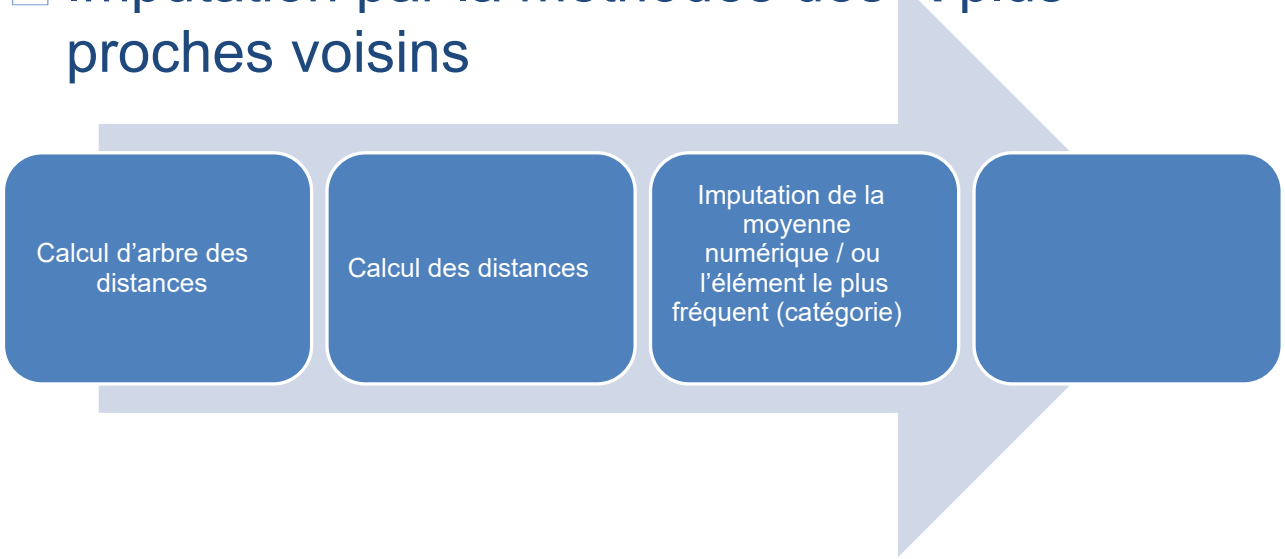


Valeurs comprises dans l'intervalle [0 g ;100 g]

Inconvénient majeur : nombre d'outliers dépendant de la taille du jeu de données
*Alternative : outliers via distance à la moyenne supérieure à 2*std*

2. Nettoyage effectué - détail

- **Traitement des NaN**
  - Imputation par la méthodes des K plus proches voisins



2. Nettoyage effectué – bilan avant/après

- 320772 lignes réduites à 77689 lignes
- 165 colonnes réduites à 40 colonnes
- 75 % de NaN réduit à 40 %
- Fichier .csv passé de 1.7 Go à 193.6 Mo

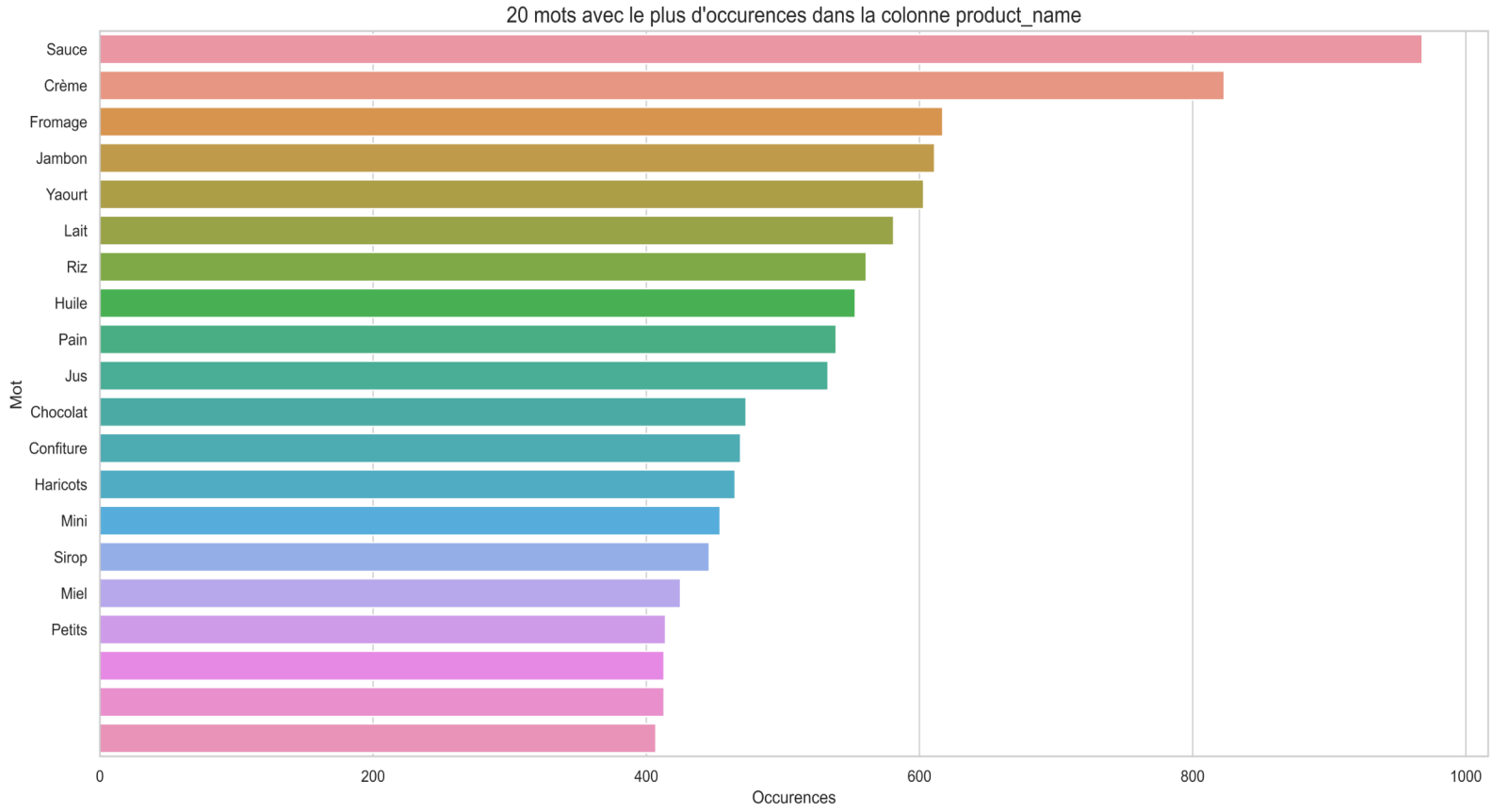
3. Analyse exploratoire – Connaissance des données :

Définition de certains champs :

- additives_n : nombre d'additifs alimentaires présents dans le produit.
- ingredients_from_palm_oil_n : nombre d'ingrédients provenant de l'huile de palme présents dans le produit.
- ingredients_that_may_be_from_palm_oil_n : nombre d'ingrédients qui peuvent provenir de l'huile de palme présents dans le produit.
- energy_100g : quantité d'énergie en kilojoules (kJ) pour 100g de produit.
- fat_100g : quantité de matières grasses en grammes (g) pour 100g de produit.
- saturated-fat_100g : quantité d'acides gras saturés en grammes (g) pour 100g de produit.
- carbohydrates_100g : quantité de glucides en grammes (g) pour 100g de produit.
- sugars_100g : quantité de sucres en grammes (g) pour 100g de produit.
- fiber_100g : quantité de fibres alimentaires en grammes (g) pour 100g de produit.
- proteins_100g : quantité de protéines en grammes (g) pour 100g de produit.
- salt_100g : quantité de sel en grammes (g) pour 100g de produit.
- sodium_100g : quantité de sodium en grammes (g) pour 100g de produit.
- Le nutrition-score-fr_100g est un score nutritionnel qui a été développé pour évaluer la qualité nutritionnelle des aliments. Ce score prend en compte différents nutriments tels que les acides gras saturés, les sucres, le sodium, les fibres et les protéines.
- Le nutrition_grade_fr est une notation nutritionnelle qui va de A à E. Elle permet de classer les aliments en fonction de leur qualité nutritionnelle, A étant la meilleure note et E la moins bonne. Cette notation est basée sur le score nutritionnel mentionné ci-dessus.

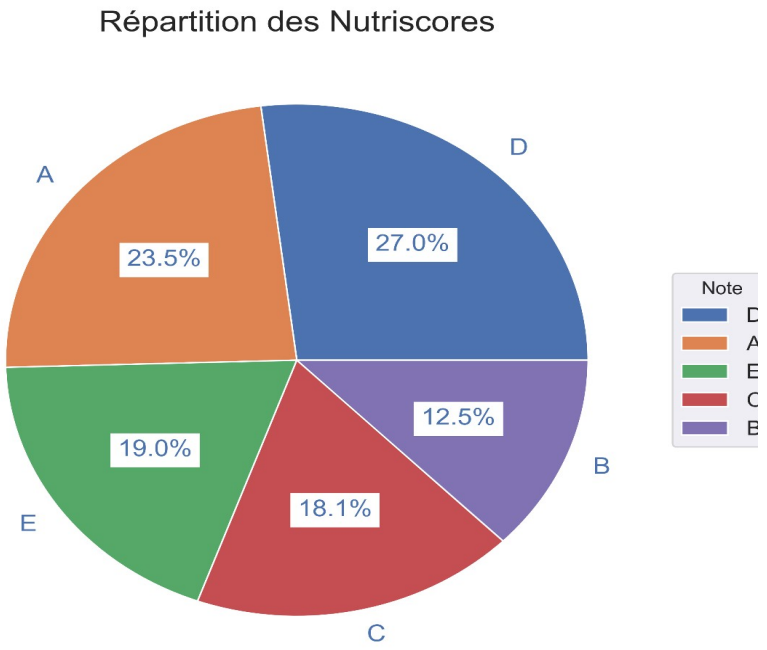
3. Analyse exploratoire – Connaissance des données :

Occurrence des mots dans les noms des produits



3. Analyse exploratoire – Connaissance des données

Répartition des nutriscores

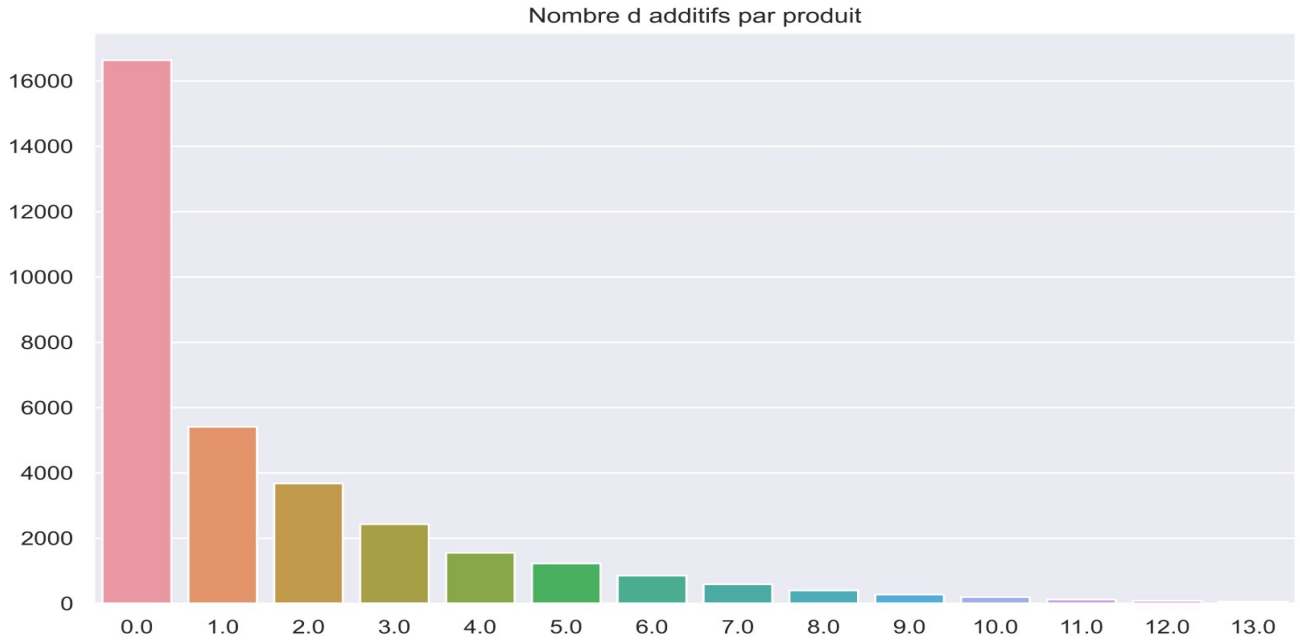



```
plt.title('Répartition des Nutriscores', size=20)
wedges, texts, autotexts = plt.pie(data.nutrition_grade_fr.value_counts().values,
        labels = data.nutrition_grade_fr.value_counts().index.str.upper(),
       autopct='%1.1f%%', textprops={'fontsize': 16,
                                     'color' : 'B',
                                     'backgroundcolor' : 'W',}
```

3. Analyse exploratoire – Connaissance des données

Additifs

```
plt.title('Nombre d\'additifs par produit')
sns.barplot(x = data.additives_n.value_counts().index,
            y = data.additives_n.value_counts().values )
```



3. Analyse exploratoire – Connaissance des données

Provenance des URL

```
data.url.dropna().apply(lambda x: str(x).split('/')[2]).unique()
```

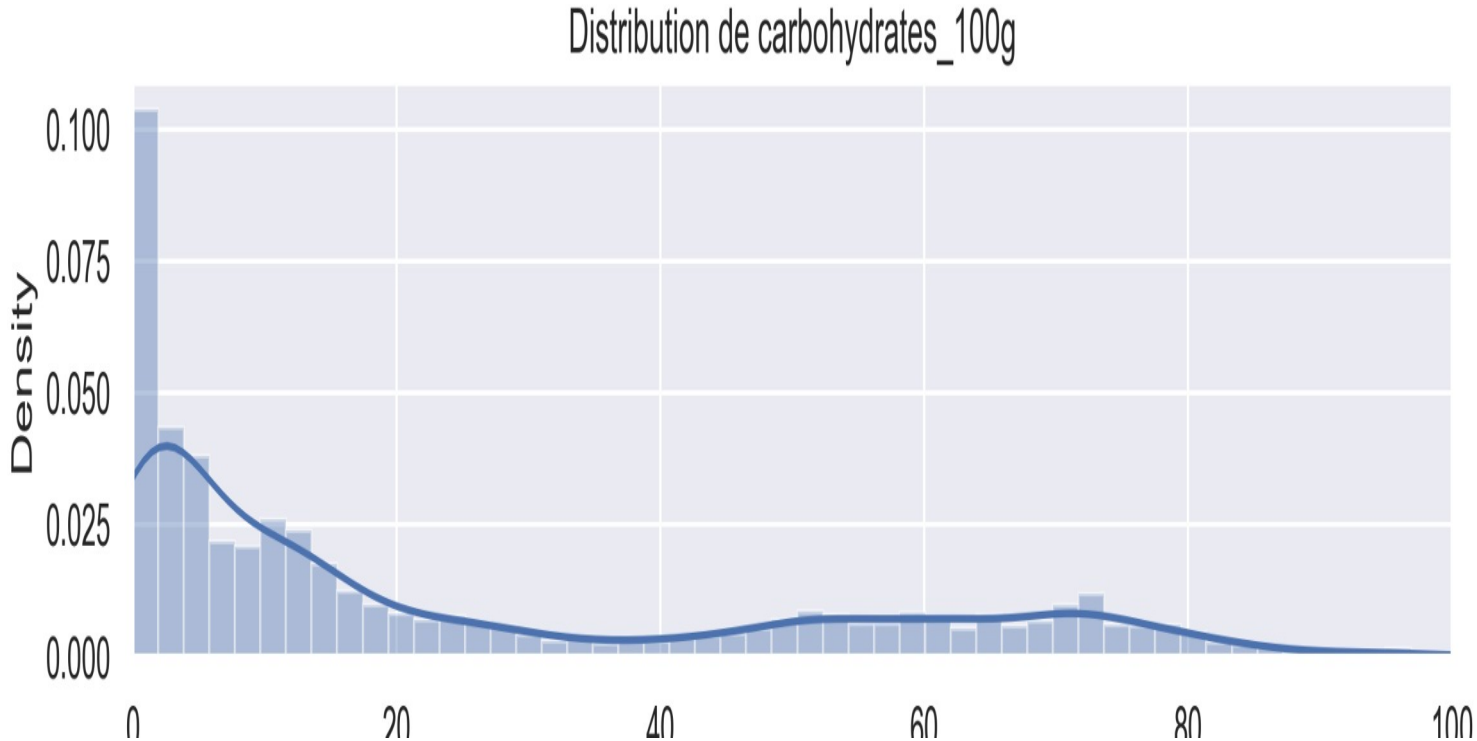
```
array(['world-en.openfoodfacts.org'], dtype=object)
```

Métriques des données numériques

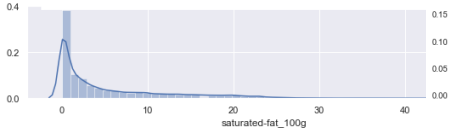
Description des variables

	energy_100g	fat_100g	saturated-fat_100g	carbohydrates_100g
count	44419.0	27828.0	42346.0	27194.0
mean	1069.14	12.77	4.96	24.69
std	756.26	17.74	7.49	26.74
min	0.0	0.0	0.0	0.0
25%	402.0	1.0	0.2	3.0
50%	1000.0	5.0	1.5	12.0
75%	1572.0	20.0	6.8	49.0
max	3700.0	100.0	56.0	96.9

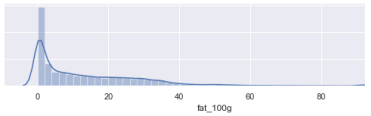
3. Analyse exploratoire – Analyse univariée : Distributions



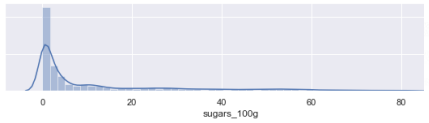
Saturated-fat_100g



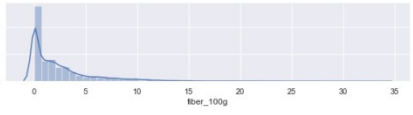
Fat_100g



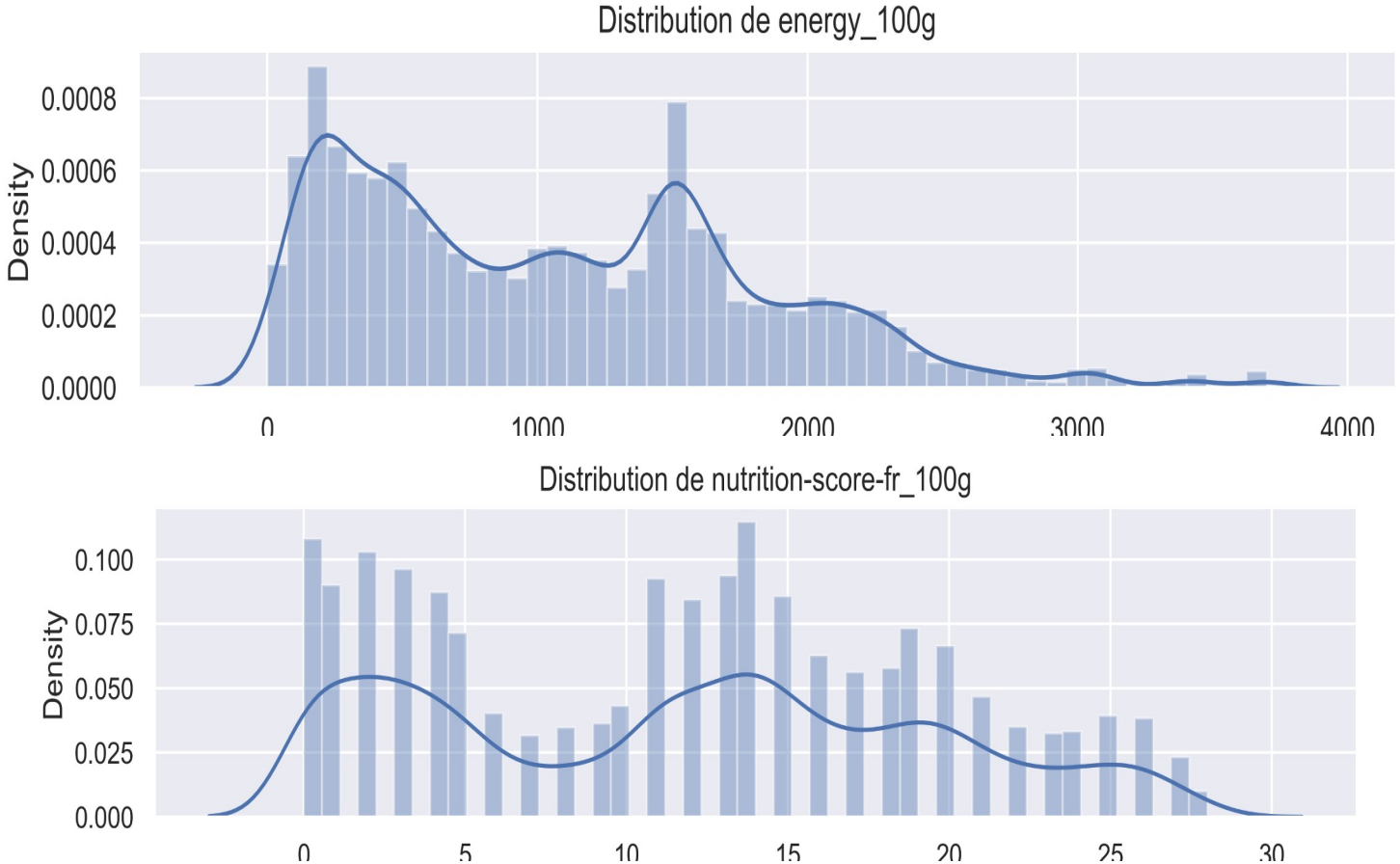
Sugars_100g



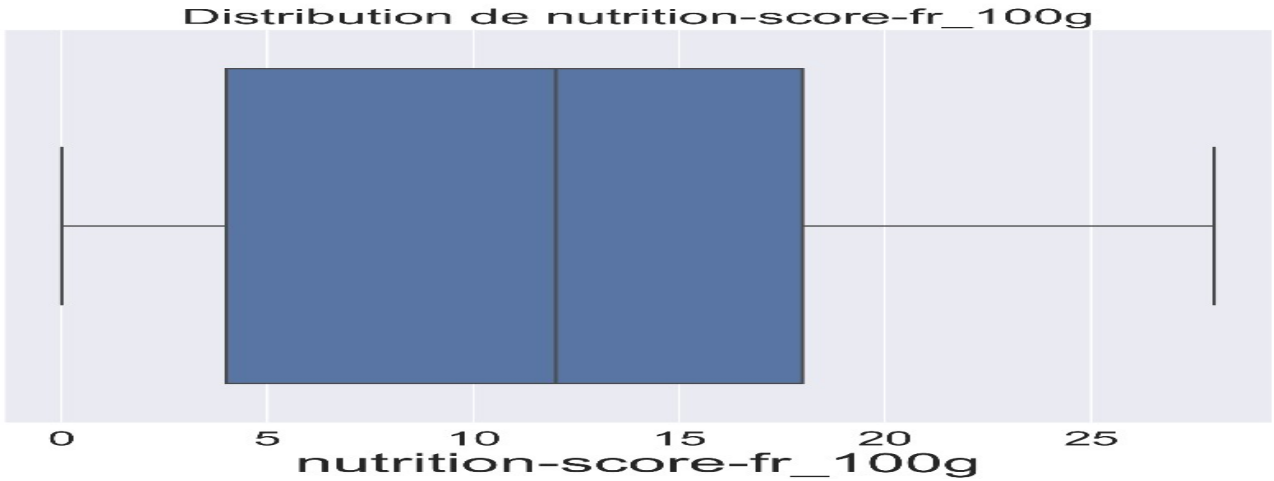
Fiber_100g

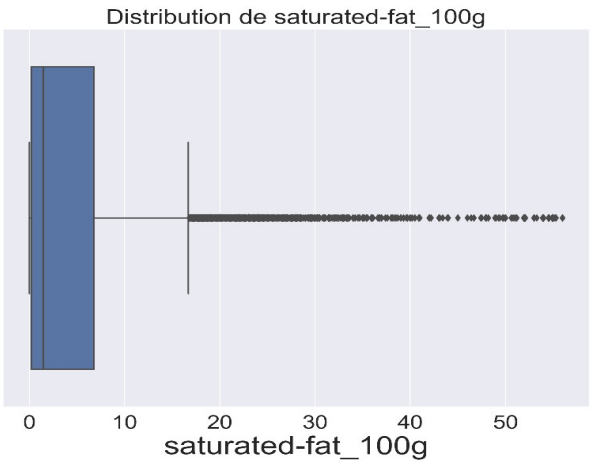


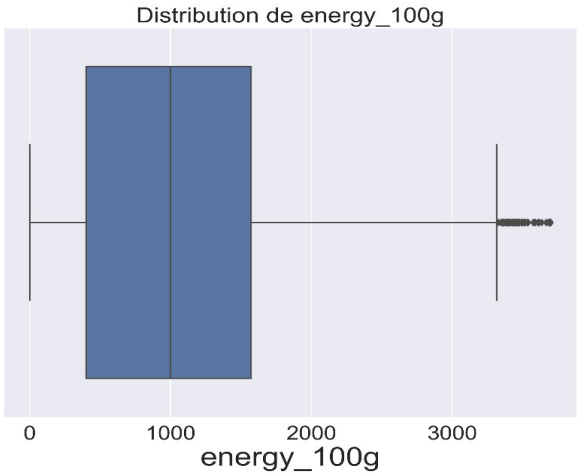
3. Analyse exploratoire – Analyse univariée : Distributions



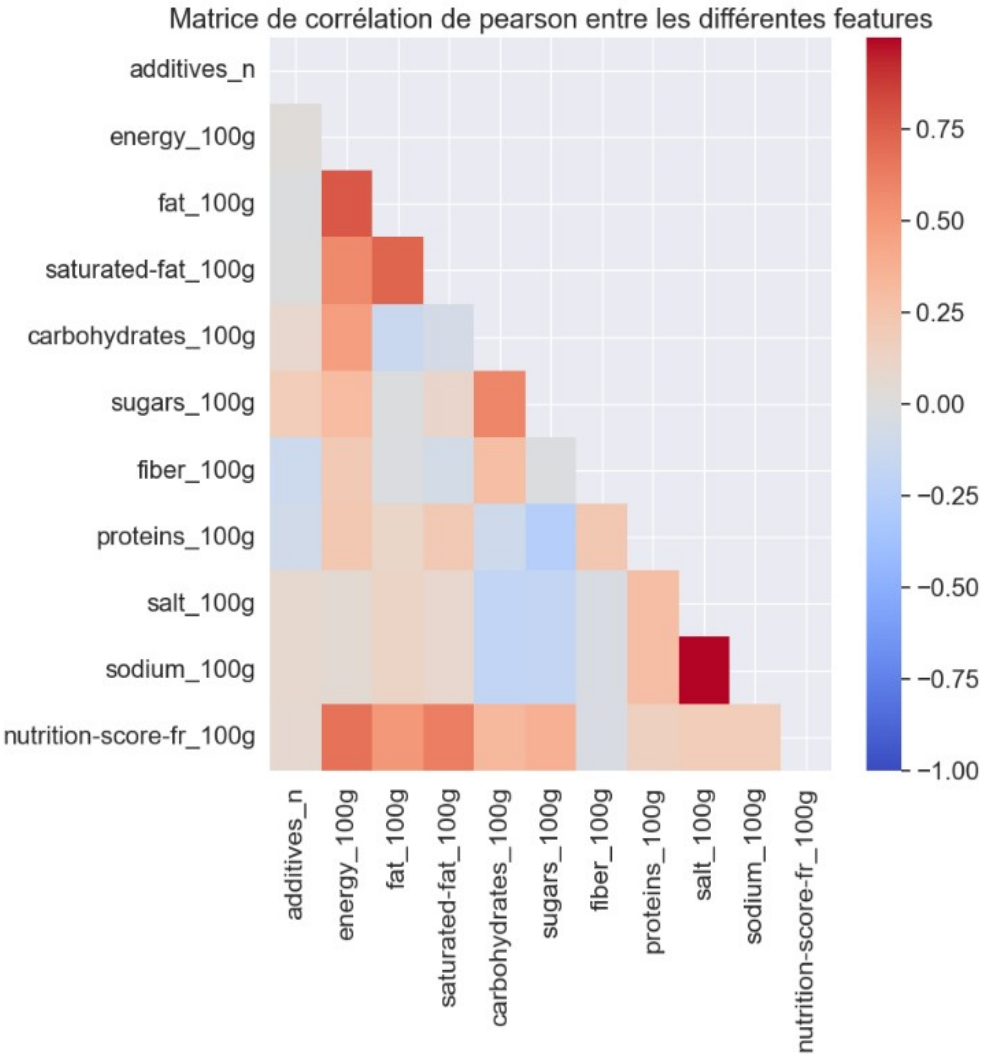
3. Analyse exploratoire – Analyse univariée : Exemple





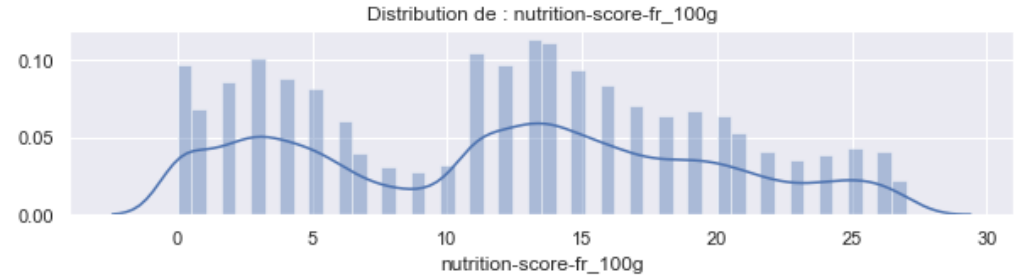


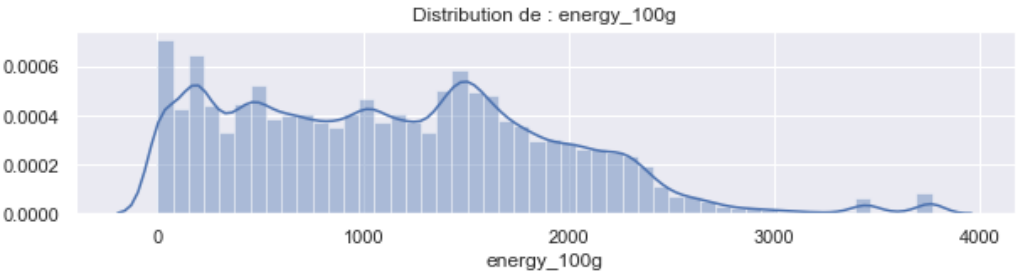
3. Analyse multivariée - corrélations

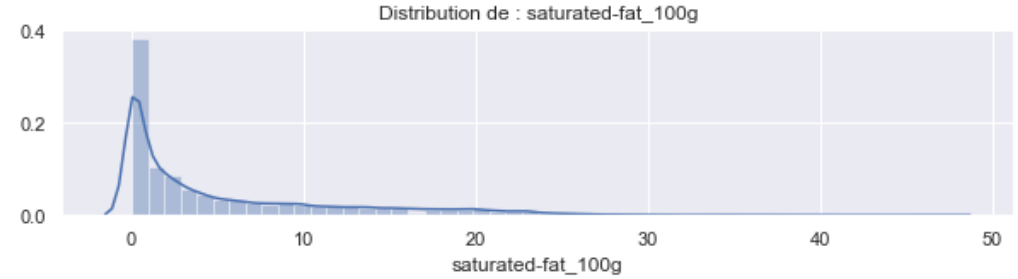


- **additives_n :** pas de correlation remarquable
- **energy_100g :** forte corrélation avec :
  - fat_100g
  - saturated-fat_100g
  - carbohydrates_100g
  - nutrition-score-fr_100g
- **fat_100g** et saturated-fat_100g fortement corrélés
- **sugars_100g** : forte correlation avec carbohydrates_100g
- **sodium_100g** correlation très forte avec salt_100g
- **nutrition-score-fr**_100g : forte corrélation avec:
  - energy_100g
  - saturated_fat_100g

3. Analyse multivariée - corrélations







- **additives_n :** pas de correlation remarquable
- **energy_100g :** forte corerlation avec :
  - fat_100g
  - saturated-fat_100g
  - carbohydrates_100g
  - nutrition-score-fr_100g
- **fat_100g** et saturated-fat_100g fortement corrélés
- **sugars_100g** : forte correlation avec carbohydrates_100g
- **sodium_100g** correlation très forte avec salt_100g
- **nutrition-score-fr**_100g : forte corrélation avec:
  - energy_100g
  - saturated_fat_100g

3. Analyse multivariée – indépendance des variables

Chi2 (χ^2) : Le test du chi2 est une méthode statistique utilisée pour évaluer si deux variables sont indépendantes ou non.Le résultat du test est une mesure du degré d'écart entre les données observées et les données théoriques.

p-value (p) : La p-value est une mesure de la probabilité de trouver des résultats aussi extrêmes que ceux observés, supposant que l'hypothèse nulle soit vraie. Elle est utilisée pour évaluer si les résultats d'une expérience sont statistiquement significatifs ou non.

Degrees of Freedom (dof) : Le nombre de degrés de liberté est un paramètre qui est utilisé dans la distribution du chi2. Il correspond au nombre de variables indépendantes dans l'analyse moins 1. Plus il y a de degrés de liberté, plus la distribution du chi2 se rapproche de la distribution normale.

En général, on considère que si la p-value est inférieure au niveau de signification (généralement fixé à 0,05), on peut rejeter l'hypothèse nulle et considérer que les deux variables sont dépendantes. Le chi2 et dof sont utilisés pour calculer la p-value.

Pour effectuer un test du Chi-2, il est nécessaire d'avoir des variables catégorielles ou discrètes, ainsi qu'un tableau de contingence qui montre la fréquence de chaque catégorie de chaque variable croisée.

La catégorisation des variables discrètes est souvent réalisée à l'aide de la fonction pd.cut de la bibliothèque Pandas en regroupant les données en catégories. Ensuite, le tableau de contingence est créé en utilisant la fonction pd.crosstab qui croise les catégories des différentes variables et compte le nombre d'observations dans chaque cellule.

Enfin, le test du Chi-2 est réalisé à partir du tableau de contingence en utilisant la fonction chi2_contingency de la bibliothèque Scipy qui calcule la statistique du Chi-2 et la p-value correspondante.

Afin d'effectuer le test chi 2, le type des variable a été modifié :

```
y = data[column].astype('category')
```

3. Analyse multivariée – indépendance des variables

Application du test du KHI2

```
#print('tableau de contingence :\n', pd.crosstab(serie1.array, serie2.array))
tab_contingence = pd.crosstab(serie1.array, serie2.array)
stat_chi2, p, dof, expected_table = chi2_contingency(tab_contingence.values)
print('chi2 : {},\np : {},\ndof : {}\n'.format(stat_chi2, p, dof))
#print('tableau de contingence : \n', tab_contingence)
```

Résultats

```
test d'indépendance nutriscore / fiber_100g
chi2 : 963.2662977794705,
p : 2.5328509986476587e-56,
dof : 361

Variables non indépendantes (H0 rejetée) car p = 2.5328509986476587e-56 <= alpha = 0.03
```

Conclusions

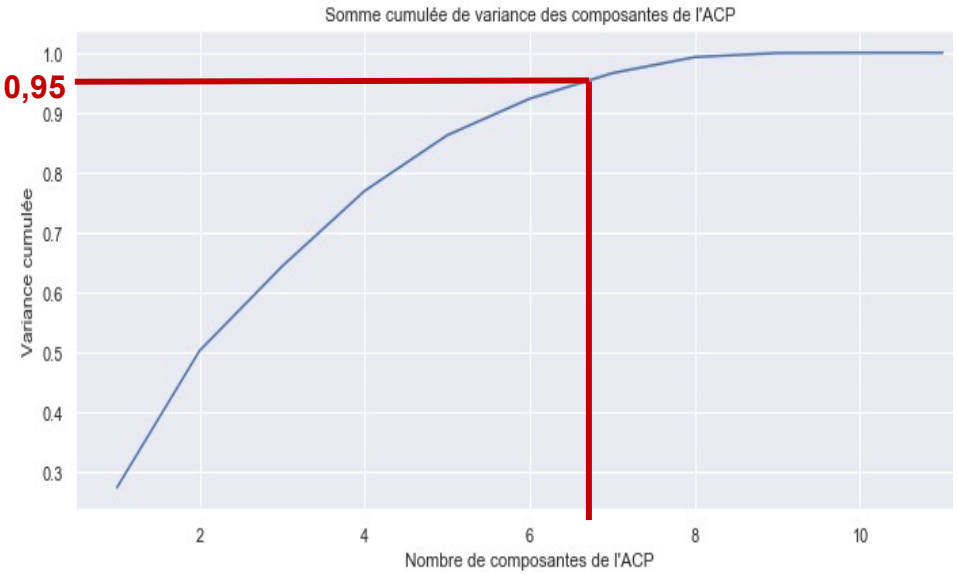
Le test d'indépendance CHI2 montre des résultats significatifs pour les deux variables étudiées avec des valeurs de chi2 très élevées et une p-value égale à 0, ce qui signifie qu'on peut rejeter l'hypothèse nulle d'indépendance entre les variables. En d'autres termes, il y a un lien significatif entre la valeur du nutriscore et la valeur de ces variables, ce qui peut aider à mieux comprendre les relations entre les différentes variables et améliorer la qualité de l'analyse des données.
Le Degré de liberté (dof) est une mesure du nombre de variables qui sont libres de varier lorsqu'on effectue un test statistique. Dans ce cas-ci, le dof est très élevé, ce qui est cohérent avec le fait que nous avons de nombreux échantillons dans notre ensemble de données. Cela signifie que les résultats du test sont plus fiables et précis.

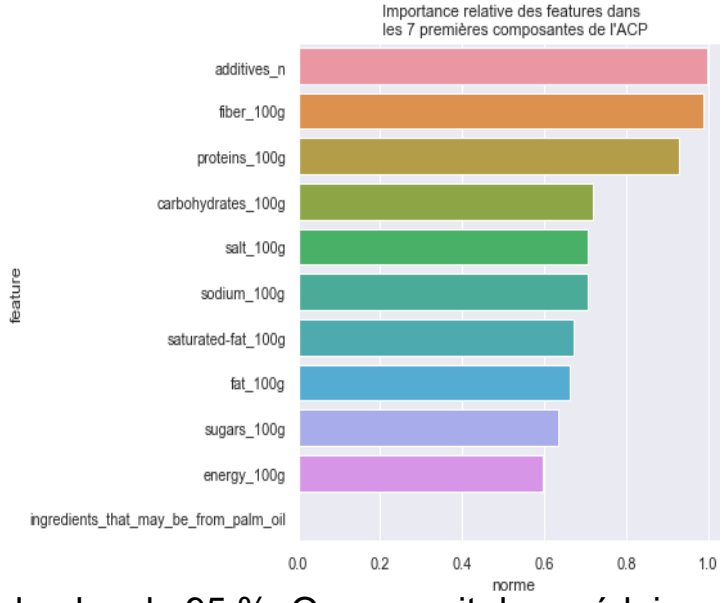
Le test du CHI2 conclut au rejet de l'hypothèse d'indépendance pour toutes nos variables:
on peut donc conclure qu'il y a un lien entre la valeur du nutriscore et la valeur des variables

3. Réduction de dimension par Analyse par Composantes Principales

```
scaler = StandardScaler()
data_pca = scaler.fit_transform(data_pca)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
```

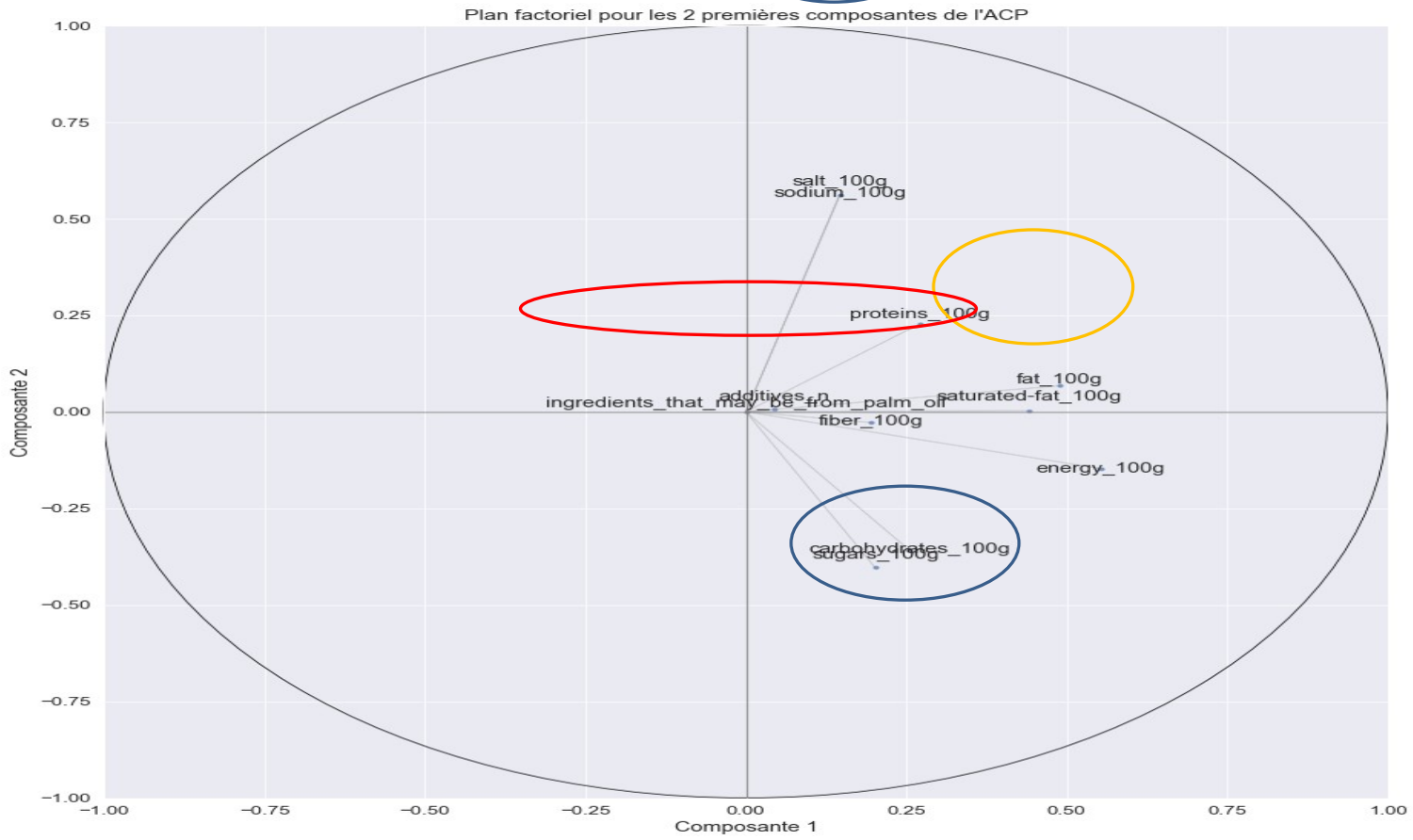
En résumé, l'ACP est une méthode qui permet de transformer un ensemble complexe de données en un ensemble de variables plus simples et compréhensibles, tout en préservant autant que possible l'information contenue dans les données initiales
L'ACP, a deux objectifs principaux, permet d'étudier :
la variabilité entre les individus, c'est-à-dire quelles sont les différences et les ressemblances entre individus ;
les liaisons entre les variables : y a-t-il des groupes de variables très corrélées entre elles, qui peuvent être regroupées en de nouvelles variables synthétiques ?





On voit qu'à partir de 7 features on a une variance cumulée de plus de 95 %. On pourrait donc réduire notre jeu de données à 7 dimensions si on souhaitait gagner en temps de calcul / volume de données.

3. Réduction de dimension par Analyse par Composantes Principales



4. Faits pertinents pour l'application

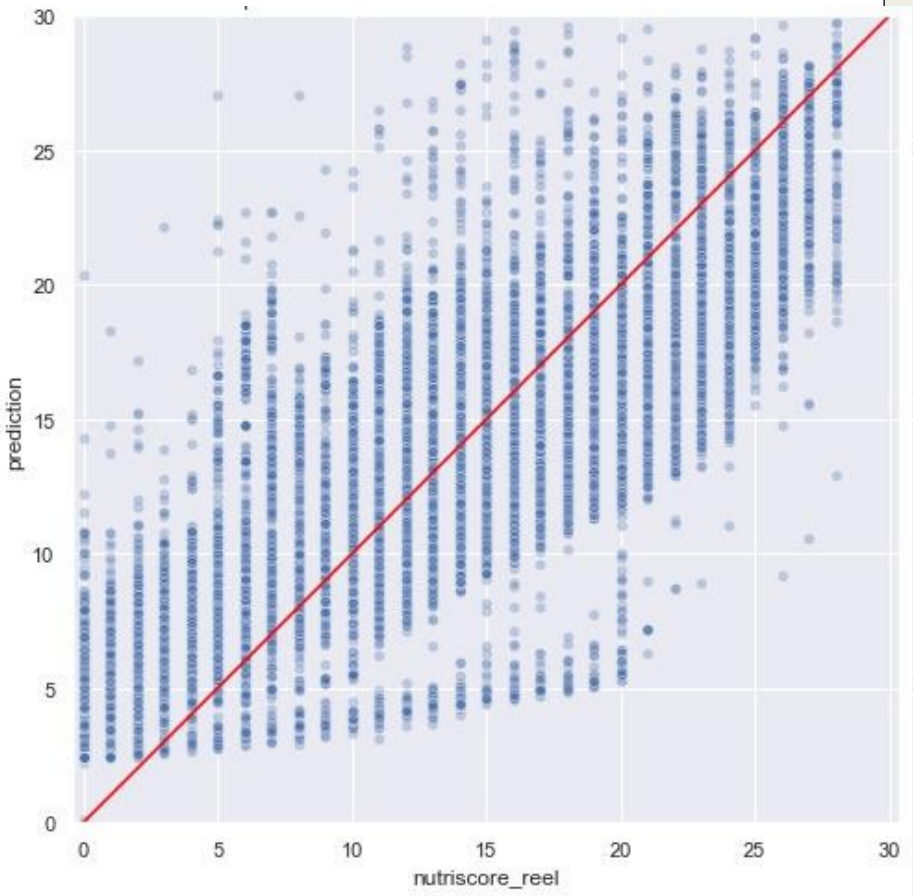
3 observations :

- Non indépendance des données
- Corrélation forte de certaines variables avec le nutriscore
- Résultats des premières régressions

4. Faits pertinents pour l'application - suite

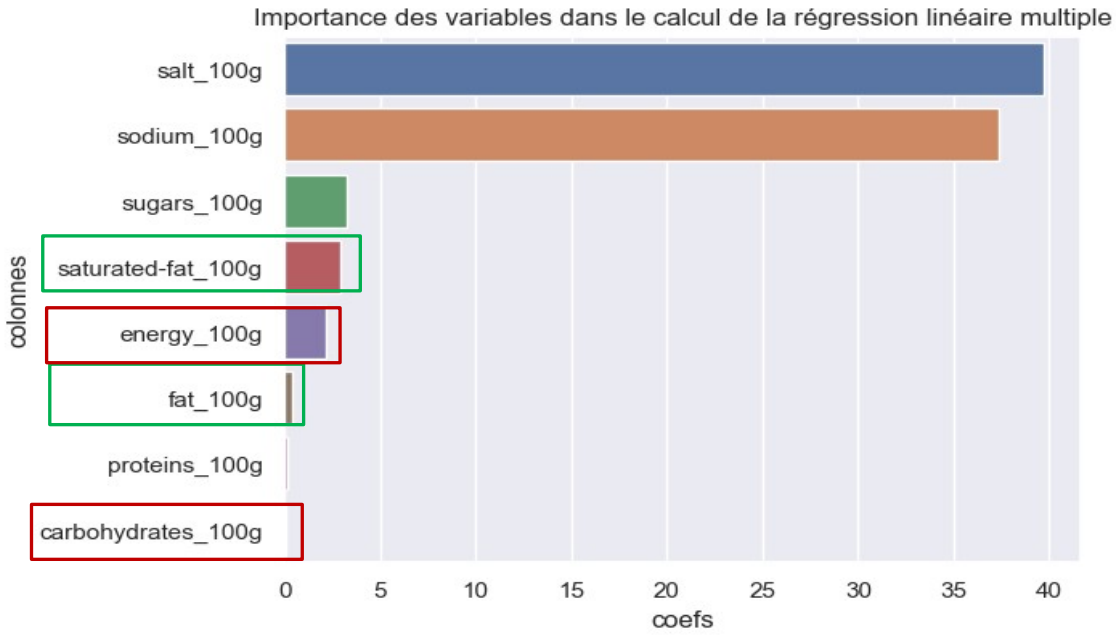
- Premières régressions
  - Jeu de données réduit à 9 variables
  - Séparation X/y
  - Séparation entrainement / test
  - Standard Scaler (μ = 0 / σ = 1)
  - Modèles : linéaire / Ridge / Lasso / Elasticnet
  - Mesure : RMSE
  - Optimisation des paramètres (Ridge / Lasso)
  - R^2 = 0,68 (Linéaire / Ridge / Lasso / Elasticnet)
  - RMSE ≈ 4,5
    (NB : nutriscore compris entre 0 et 26)

Comparaison des nutriscores estimés (y) et réels (x)



4. Faits pertinents pour l'application - suite

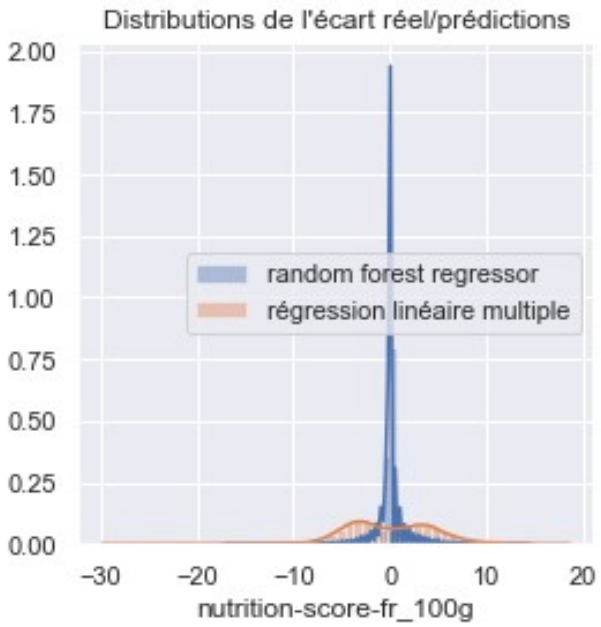
- Vérification : comparaison de l'importance des variables avec corrélation de Pearson

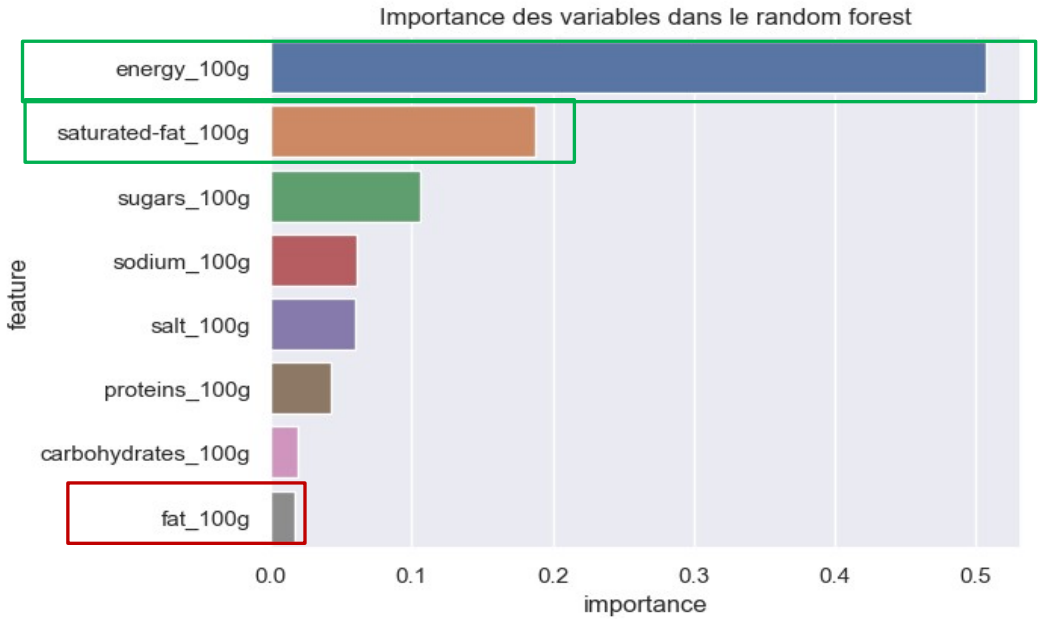


NB : La Forte corrélation de fat_100g et energy_100g avec nutrition-score ne se retrouve pas ici

4. Faits pertinents pour l'application - suite

- Application d'un algorithme d'ensemble (forêts aléatoires)
  - *R^2 = 0,94 sur le jeu de données de test*
  - *RMSE = 1,85 (progrès d'un facteur 2,5 x)*





5. Synthèse

- Forêts aléatoires concluantes sur un jeu de données réduit *(NB : Vrai algorithme du nutriscore non linéaire : cohérent)*
- Résultats cohérents avec les principes nutritionels (graisses saturées, sucres)
- Possibilité de proposer un algorithme qui donne une indication de nutriscore approché ($R^2 = 0,85$) avec 8 variables : faisabilité de l'application

Merci de votre attention