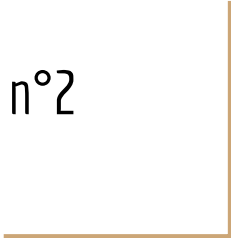


Cursus Ingénieur Machine Learning

-
Vincent Jugé

-
Soutenance Projet n°2

-
02/12/2021





Application Proposée

Qu'est ce qu'il y a *-de bon-* au menu ?



Contexte

Partant des constats que :

- Le nutriscore est attribué à chaque produit
- Une note 'E' ne veut pas dire qu'il faut bannir ce produit, mais qu'il faut *raisonner* sa consommation
- A l'inverse, il ne faut pas se nourrir exclusivement de produits notés 'A'

Problématique

- Comment peut faire le consommateur pour préparer des **menus équilibrés** ?
- **Quels produits** choisir et en **quelle quantité** ?

Le nutriscore d'un seul produit n'est pas suffisant

= besoin d'un **score par menu**

Value Proposition

- Apporter un nutriscore relatif à un repas (entrée, plat, dessert)
- Le “MenuScore” est l'équivalent du nutriscore mais pour un menu complet
 - Plat = somme des quantités de nutriments
 - MenuScore = Somme(Plats / nb de personnes)
- Ceci revient à calculer un produit fictif (le menu complet), et à calculer son nutriscore

Or, nous ne connaissons pas l'algorithme pour ce calcul !

Objectif : trouver un modèle qui se base sur les observations de la base open food facts

Menu - Exemple

Carottes râpées en sauce

~

Couscous Merguez

~

Fondants chocolat

	nutriscore_grade	nutriscore_score	product_name	energy-kcal_100g	energy_100g	fat_100g	saturated-fat_100g	carbohydrates_100g	sugars_100g	proteins_100g	salt_100g	sodium_100g
468699	c	3.0	Fondant chocolat	136.0	569.0	5.5	3.6	17.3	16.2	3.5	0.14	0.056
648372	d	18.0	merguez	255.0	1067.0	21.0	9.0	2.0	1.0	14.0	1.00	0.720
667250	b	0.0	Légumes couscous	28.0	117.0	0.7	0.0	3.3	1.8	1.2	0.55	0.220
798108	a	-1.0	Carottes râpées en sauce	102.0	421.0	8.2	1.1	5.1	4.7	0.7	0.81	0.324
951017	a	-1.0	semoule	370.0	1548.0	2.0	0.4	73.0	3.0	13.2	0.05	0.020

Chargement et nettoyage du dataset

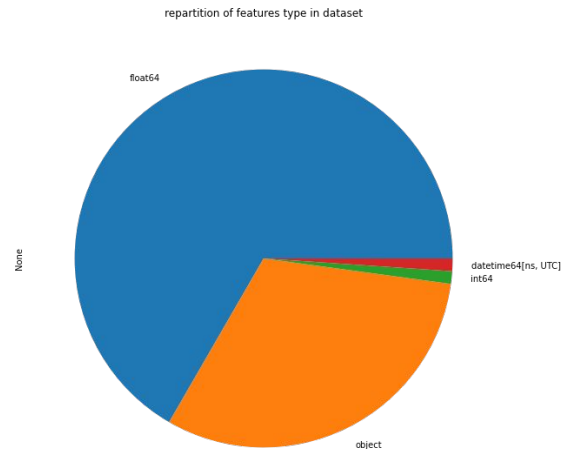
Découverte du dataset

Le dataset contient 186 features

4 types de données

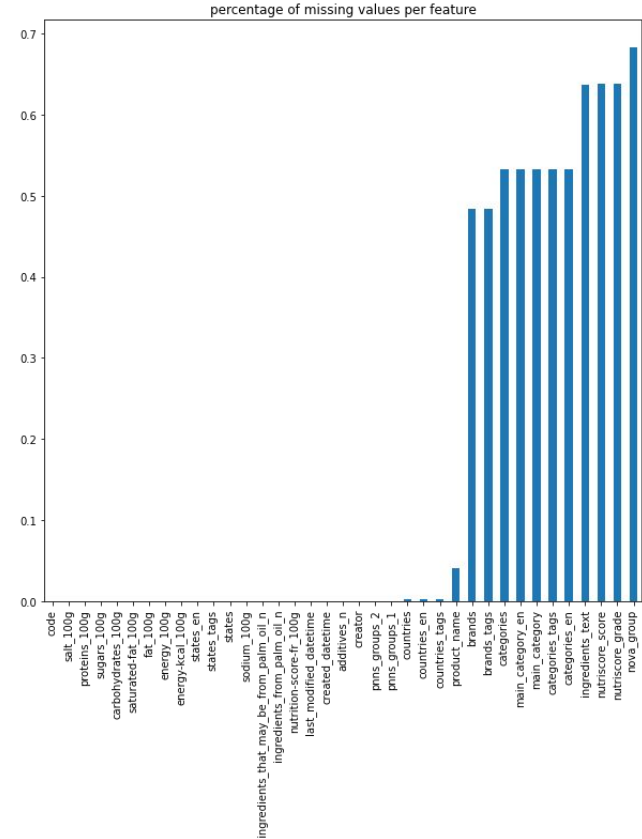
1'988'476 valeurs

* Pour réduire le temps de chargement en mémoire (lecture csv + parsing), on utilise la librairie *Modin* qui permet d'utiliser tous les cores cpu et expose la même API que *Panda*



Chargement et Nettoyage

- On s'aperçoit que beaucoup de features sont vides.
- On positionne un seuil à 70% maximum de données manquantes, autrement on enlève la feature
- Pour les données quantitatives, on applique :
 - un remplacement des valeurs manquantes par 0.0
 - on enlève les données aberrantes (> 100 pour 100g)
- On supprime les doublons (très peu)



Contenu final du dataset

Après nettoyage, le dataset contient 37 features, de 3 types : object, float, date.
Décrivant des features qualitatives et quantitatives.

Qualitative Features:

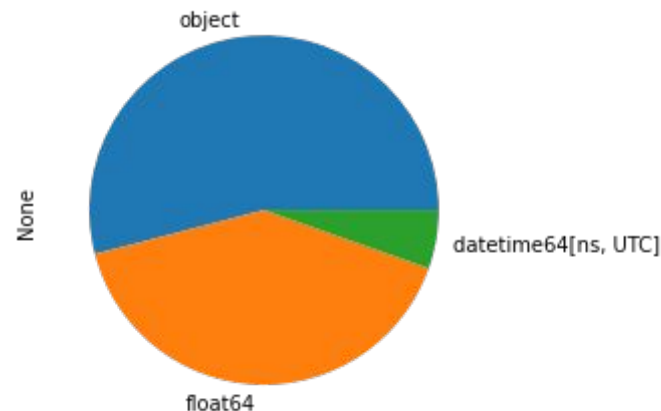
`categories_tags`, `states`, `created_datetime`, `code`, `brands`, `product name`,
`last modified datetime`, `creator`, `brands_tags`, `countries`, `categories`, `ingredients_text`,
`countries_en`, `categories_en`, `states_tags`, `main_category`, `states_en`, `main_category_en`,
`countries_tags`

Qualitative ordinal Features:

`additives_n`, `ingredients_from_palm_oil_n`, `ingredients_that_may_be_from_palm_oil_n`,
`nutriscore_grade`, `nova_group`, `pnns_groups_1`, `pnns_groups_2`

Quantitative Features:

`nutriscore_score`, `energy-kcal_100g`, `energy_100g`, `fat_100g`, `saturated-fat_100g`,
`carbohydrates_100g`, `sugars_100g`, `proteins_100g`, `salt_100g`, `sodium_100g`,
`nutrition-score-fr_100g`



Analyses

Outillage

Pour effectuer certaines analyses, on se dote de fonctions, cf. fichier `module_P2_utils.ipynb`

Par exemple, `agg_func`, qui permet d'afficher dans un même tableau : écart-type, skew, kurtosis, moyenne, mediane, variance, mad, produit, somme

	std	skew	kurtosis	mean	median	var	mad
additives_n	2.00	4.17	24.10		0.74	0.00	4.01
ingredients_from_palm_oil_n	0.09	11.98	148.37		0.01	0.00	0.01
ingredients_that_may_be_from_palm_oil_n	0.19	9.15	108.20		0.03	0.00	0.03
nutriscore_score	8.84	0.10	-0.94		9.10	10.00	78.16
nova_group	0.97	-1.65	1.43		3.42	4.00	0.95
energy_kcal_100g	6,168,468,420.38	1,409.26	1,986,208.99	4,421,061.51	159.00	38,050,002,653,244,399,616.00	
energy_100g	4,729,104,920,095,283,981,120,154,818,7...	1,409.48	1,986,622.00	3,355,222,527,933,793,906,480,431,211,1...	715.00	22,364,433,345,269,419,728,237,164,119,...	6,710,441,678,050,853,204,8
fat_100g	16.66	2.47	7.79		10.78	2.68	277.61
saturated-fat_100g	7.15	3.82	26.62		3.89	0.50	51.12
carbohydrates_100g	27.35	1.03	-0.32		22.14	7.00	748.03
sugars_100g	17.97	2.29	5.04		10.21	1.60	323.00
proteins_100g	9.48	2.78	13.66		6.83	3.40	89.84
salt_100g	4.08	15.90	315.16		1.00	0.13	16.68
sodium_100g	1.63	15.90	315.26		0.40	0.05	2.67
nutrition-score-fr_100g	6.89	1.73	2.05		3.29	0.00	47.40

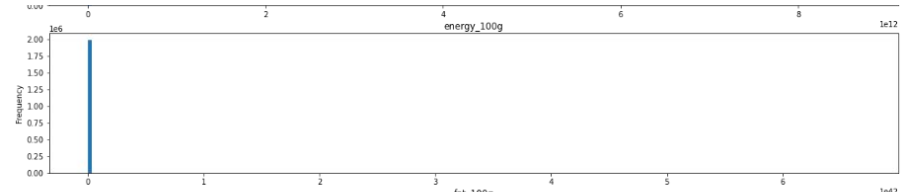
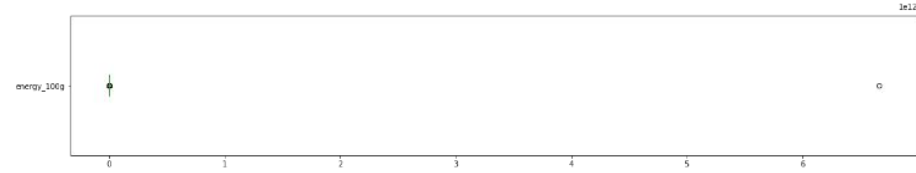
Analyse Quantitative

On procède à une analyse de la variance et de la distribution des données

→ on constate que des outliers sont présents et qu'il convient de les supprimer pour obtenir des features utilisables dans une analyse

exemple ci contre d'une seule feature 'energy'

L'analyse est peu probable car la distribution est très resserrée, trop d'outliers sont présents



Analyse Quantitative - Supprimer les Outliers

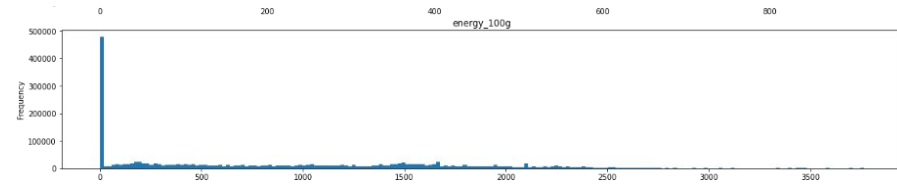
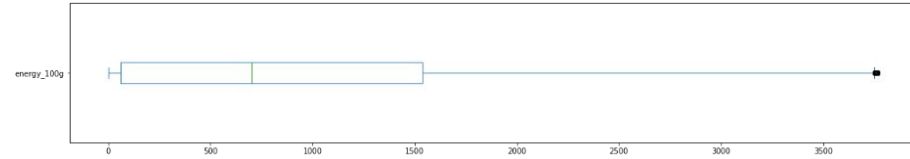
On applique la méthode *IQR* :

- on calcule les quantiles Q1 (25%) et Q3 (75%)
- $IQR = Q3 - Q1$
- On supprime les valeurs comprises entre:

$$Q1 - 1.5 * IQR < \text{valeur} < Q3 + 1.5 * IQR$$

* Note: une autre méthode pourrait s'appliquer, celle du z-score

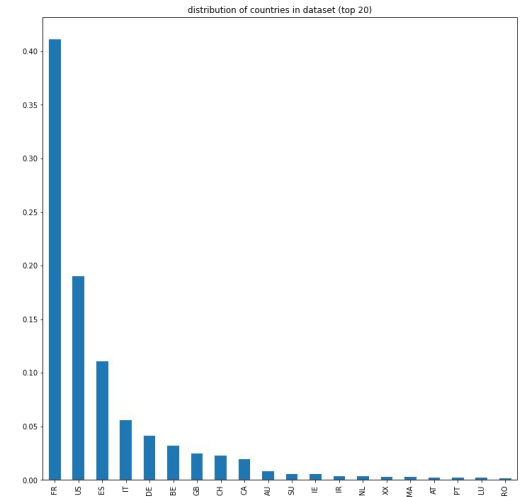
Après suppression des outliers, on obtient des valeurs de distribution plus intéressantes



*Exemple feature 'energy_100g'
Boxplot et dispersion après suppression des outliers*

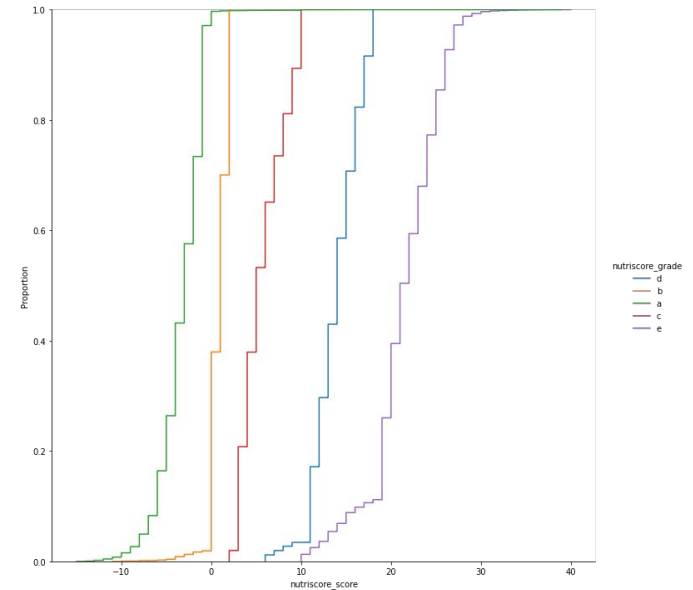
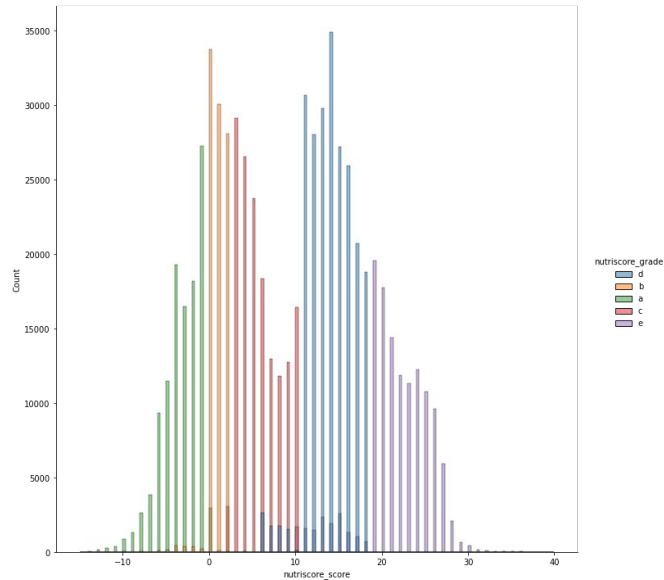
Analyse Qualitative

- On cherche à connaître les modalités, et le mode pour chaque feature.
- On applique une catégorisation, par exemple à la feature 'countries',
 - Elle contient beaucoup de modalités, qui manifestement représentent la même information (France, en:FR, en:fr, ...)
 - On arrive à constater que le dataset est composé principalement de produits provenant de 4 pays, avec une grande majorité pour la France



Analyse multivariée & corrélations

- On s'aperçoit que le 'nutriscore_score' et 'nutriscore_grade' sont corrélés, mais pas de façon linéaire
- De plus, les deux features se "recouvrent"

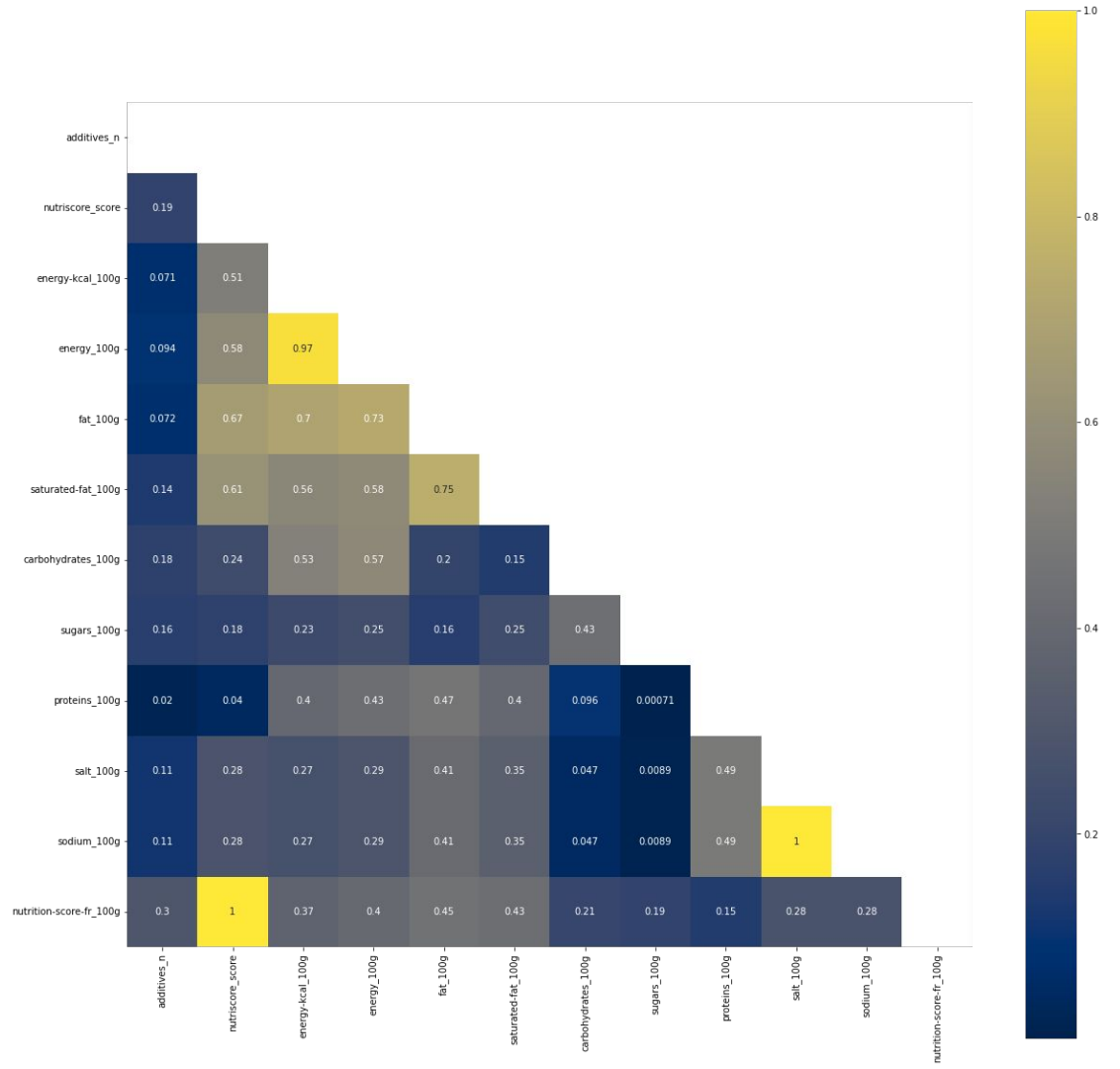


Analyse multivariée & corrélations

- On calcul de coefficient de corrélation (valeur absolue, méthode de Pearson) entre chaque feature quantitative
- On les représente
 - soit dans un tableau de contingence
 - soit sur une matrice (heatmap)
- On considère qu'une corrélation existe entre deux variable si le score est > 0.5
- Dans ce cas on devrait pouvoir proposer un modèle de prédiction basé sur celles-ci

$$\rho_{X,Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

Matrice de corrélation



Corrélation - Observations et conclusions

On observe que certaines features sont corrélées :

- energy & energy-kcal
- nutriscore_score & nutriscore-score-fr
- sodium & salt

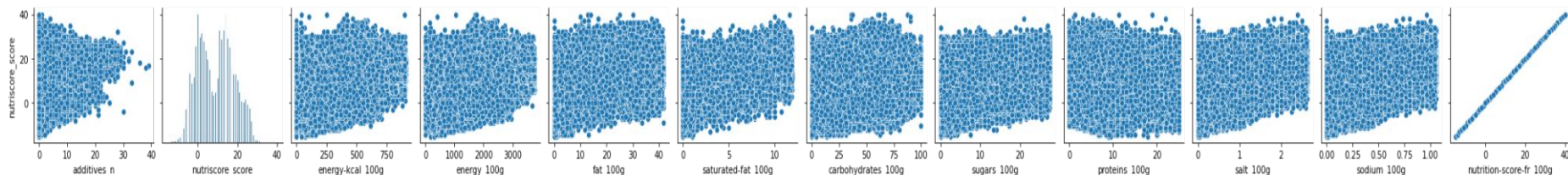
⇒ ceux là sont évidents, ne nous apporte rien. En revanche nutriscore est corrélé avec:

- saturated_fat
- fat
- energy
- energy_kcal
- nova_group

Avec ces valeurs, on devrait pouvoir obtenir de bons résultats

Corrélations du nutriscore_score

Graphiques de corrélation sur l'ensemble des features de type nutriments



On constate que des tendances linéaires se dégagent

Modèle de prédiction

Prédiction du nutriscore_score

Corrélation uni-dimensionnelle avec fat

Modèle à Régression linéaire

⇒ on obtient un score $R^2 = 0.35$

si $R^2 = 0$, le modèle n'est pas meilleur qu'une prédiction basée sur la moyenne

si $R^2 = 1$, les prédictions sont parfaites

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

```
X = df.dropna().loc[:, 'fat_100g'].values.reshape(-1, 1)
y = df.dropna().loc[:, 'nutriscore_score'].values.reshape(-1, 1)
```

```
reg = LinearRegression().fit(X, y)
```

```
reg.coef_
```

```
array([[0.47787512]])
```

Score is the R^2 defined as $(1 - \frac{u}{v})$, where u is the residual sum of squares $((y_true - y_pred)** 2).sum()$ and v is the total sum of squares $((y_true - y_true.mean()) ** 2).sum()$

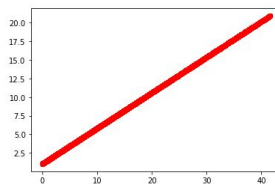
```
reg.score(X, y)
```

```
0.3501325468540978
```

In that case, R^2 is not very good

```
nutriscore_pred = reg.predict(X)
plt.scatter(df.dropna()['fat_100g'], nutriscore_pred, color='red')
```

<matplotlib.collections.PathCollection at 0x7fe9bc019be0>



We can try to make a linear regression with multiple features

Prédiction du nutriscore_score

Corrélation multi-dimensionnelle avec fat, energy, sugars, salt

Modèle à Régression linéaire

⇒ on obtient un score $R^2 > 0.5$, ce qui est bien meilleur, mais encore perfectible

```
X = df.dropna()[['fat_100g', 'energy_100g', 'sugars_100g', 'salt_100g']].to_numpy()
y = df.dropna()['nutriscore_score'].to_numpy().reshape(-1, 1)
```

```
reg = LinearRegression().fit(X, y)
```

```
reg.coef_
```

```
array([[ 3.95715190e-01, -1.66460882e-03,  4.24424202e-01,
         4.25859762e+00]])
```

```
reg.score(X, y)
```

```
0.5630929468277883
```

Here, the R^2 score is better, we are above 0.5

En injectant toutes les features nutriments dans notre modèle, on devrait pouvoir effectuer des prédictions sur le nutriscore

Corrélation du grade

- Obtenir un score est intéressant mais pas très utile pour le consommateur.
- Pour obtenir le nutriscore_grade, sachant qu'il n'est pas linéairement corrélé au nutriscore_score, un modèle de type 'classifier' serait préférable

Conclusions

Application - Étapes suivantes

Pour pouvoir prédire le menu score sous forme de lettre il faudrait:

- Avoir un modèle de prédiction du score (chiffre) plus performant, atteindre un R^2 d'au moins 0.8
 - Ajouter des variables
 - Refaire une passe de nettoyage, beaucoup de valeurs sont à 'unknown'
 - Réduire le threshold de features vides de 70% à 50% par exemple
 - 200'000 valeurs (10% du dataset d'origine) devraient pouvoir suffire à obtenir un modèle performant
- Avoir un modèle de classification pour prédire le nutriscore grade (lettre)
- Ces étapes n'ont pas été complétées car en dehors du champ de cette étude

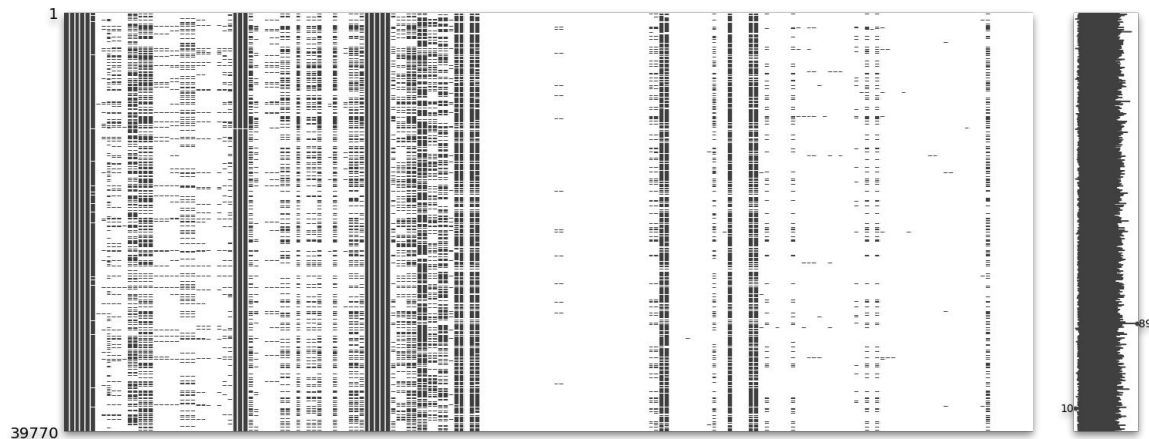
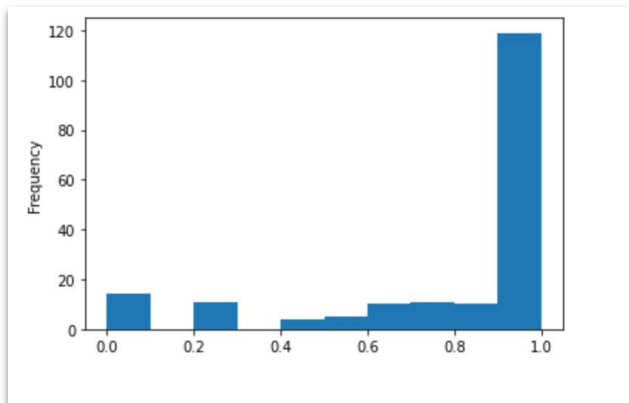
Application - Étapes suivantes

- Pour récupérer les identifiants de produits, un scan du code barre via une application sera envisagé, ou
- Si le consommateur fait ses courses via un drive, permettre via le drive de récupérer les codes barres, via API

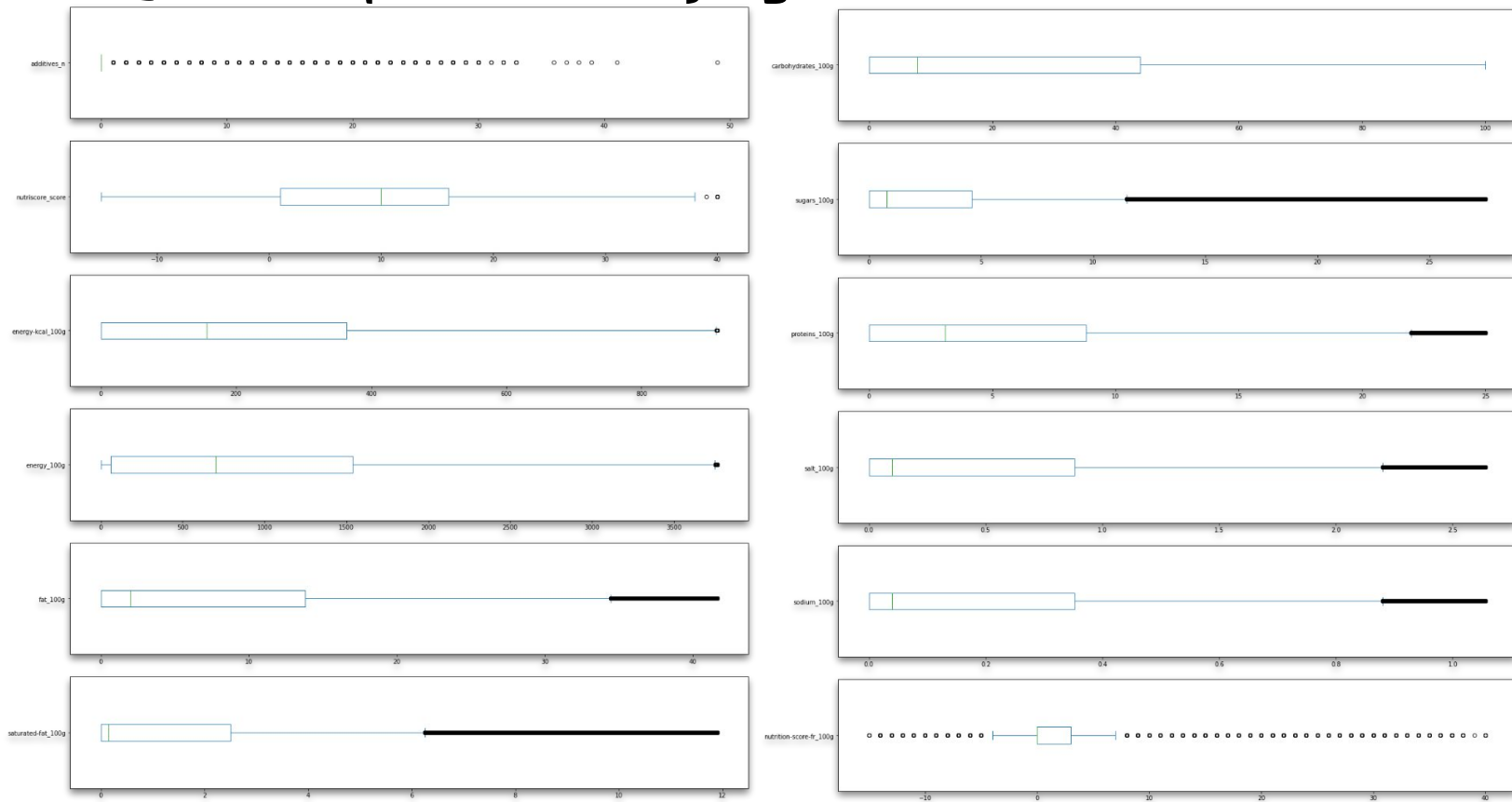
L'objectif est de proposer le MenuScore au moment où le consommateur fait ses achats, pour prévoir la qualité des menus qu'il va se préparer dans les jours suivants.

Annexes

Analyse du dataset



Analyse Quanti après nettoyage



Analyse Quanti après nettoyage

	std	skew	kurtosis	mean	median	var	mad	prod	sum
additives_n	2.00	4.17	24.10	0.74	0.00	4.01	1.17	0.00	1,465,137.00
nutriscore_score	8.84	0.10	-0.94	9.10	10.00	78.16	7.63	-0.00	6,541,544.00
energy-kcal_100g	202.93	0.76	-0.15	207.32	157.00	41,181.93	174.40	0.00	410,634,407.38
energy_100g	831.60	0.67	-0.40	884.77	703.00	691,554.32	718.21	0.00	1,746,961,011.59
fat_100g	10.86	1.29	0.48	8.04	2.00	117.89	8.89	0.00	15,182,834.34
saturated-fat_100g	2.81	1.76	2.16	1.77	0.15	7.90	2.14	0.00	3,102,651.01
carbohydrates_100g	27.35	1.03	-0.32	22.14	7.80	748.03	23.39	0.00	43,974,357.79
sugars_100g	6.28	2.03	3.40	3.86	0.80	39.49	4.53	0.00	6,599,723.18
proteins_100g	6.71	1.23	0.55	5.56	3.10	44.96	5.40	0.00	10,578,741.42
salt_100g	0.66	1.29	0.59	0.48	0.10	0.44	0.55	0.00	899,066.51
sodium_100g	0.26	1.29	0.59	0.19	0.04	0.07	0.22	0.00	359,620.97
nutrition-score-fr_100g	6.89	1.73	2.05	3.29	0.00	47.40	5.21	-0.00	6,541,651.00

Tableau de Corrélations

	additives_n	nutriscore_score	energy-kcal_100g	energy_100g	fat_100g	saturated-fat_100g	carbohydrates_100g	sugars_100g	proteins_100g	salt_100g	sodium_100g	nutrition-score-fr_100g
additives_n	1.000000	0.190459	0.071430	0.093512	0.072095	0.136921	0.178189	0.161570	0.019509	0.111630	0.111616	0.295180
nutriscore_score	0.190459	1.000000	0.514700	0.578320	0.665422	0.613796	0.242438	0.182483	0.039889	0.284026	0.284033	1.000000
energy-kcal_100g	0.071430	0.514700	1.000000	0.967918	0.704853	0.561156	0.530611	0.229331	0.397280	0.266109	0.266115	0.366878
energy_100g	0.093512	0.578320	0.967918	1.000000	0.731808	0.575508	0.567977	0.249252	0.426721	0.287247	0.287243	0.401921
fat_100g	0.072095	0.665422	0.704853	0.731808	1.000000	0.751978	0.197372	0.158036	0.466061	0.410890	0.410882	0.448845
saturated-fat_100g	0.136921	0.613796	0.561156	0.575508	0.751978	1.000000	0.148301	0.245613	0.396212	0.350178	0.350169	0.429571
carbohydrates_100g	0.178189	0.242438	0.530611	0.567977	0.197372	0.148301	1.000000	0.433564	0.095887	0.046723	0.046708	0.206928
sugars_100g	0.161570	0.182483	0.229331	0.249252	0.158036	0.245613	0.433564	1.000000	0.000709	0.008925	0.008924	0.191573
proteins_100g	0.019509	0.039889	0.397280	0.426721	0.466061	0.396212	0.095887	0.000709	1.000000	0.490704	0.490712	0.147946
salt_100g	0.111630	0.284026	0.266109	0.287247	0.410890	0.350178	0.046723	0.008925	0.490704	1.000000	0.999988	0.280556
sodium_100g	0.111616	0.284033	0.266115	0.287243	0.410882	0.350169	0.046708	0.008924	0.490712	0.999988	1.000000	0.280559
nutrition-score-fr_100g	0.295180	1.000000	0.366878	0.401921	0.448845	0.429571	0.206928	0.191573	0.147946	0.280556	0.280559	1.000000