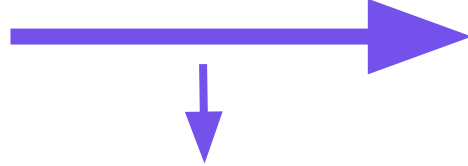


Formation DS-IML PROJET 2





APPEL À PROJET



Faisabilité d'un **MOTEUR DE RECOMMANDATION**
(étude exploratoire des données + prototype)



- 1** - Prise de contact / Nettoyage des données
- 2** - Analyses univariées (corrections)
- 3** - Analyses bivariées (corrélations entre var.)
- 4** - Utilisation d'un modèle ML (complétion de data)
- 5** - Réalisation d'un prototype de moteur de rec.
- 6** - Conclusion & Perspectives



Phase 1 : Prise de contact et premier nettoyage

Sélection des variables en deux étapes :

a) Pertinence - Quantité de données renseignées

b) Choix - Redondances

Détail de la sélection par **tranches** :

1- Infos. Gén. : 'code', 'url', '**product_name**'

2- Tags : 'countries_en', '**brands**'

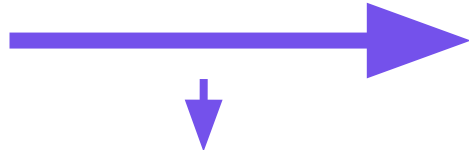
3- Misc. Ingr. : 'allergens', '**nova_group**', 'additives_en', 'additives_n', 'ingredients_from_palm_oil_n',
'ingredients_that_may_be_from_palm_oil_n', 'nutriscore_score', '**nutriscore_grade**'

4- Nutrition : 'energy-kcal_100g', 'energy_100g', 'fat_100g', 'saturated-fat_100g', 'trans-fat_100g',
'cholesterol_100g', 'carbohydrates_100g', 'sugars_100g', 'fiber_100g', 'proteins_100g',
'salt_100g', 'sodium_100g', 'vitamin-a_100g', 'vitamin-c_100g', 'calcium_100g', 'iron_100g'

181**29**



APPEL À PROJET



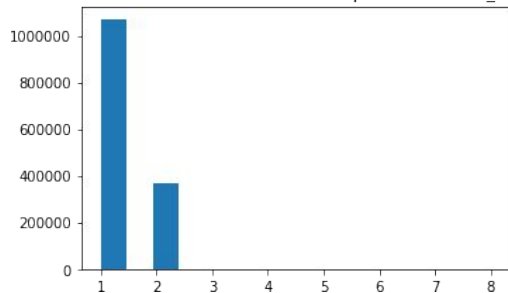
Faisabilité d'un **MOTEUR DE RECOMMANDATION**
(étude exploratoire des données + prototype)

- 1 - Prise de contact / Nettoyage des données
- 2 - Analyses univariées (corrections)
- 3 - Analyses bivariées (corrélations entre var.)
- 4 - Utilisation d'un modèle ML (complétion de data)
- 5 - Réalisation d'un prototype de moteur de rec.
- 6 - Conclusion & Perspectives

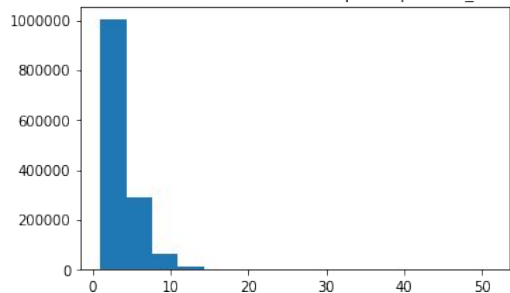
Phase 2 : Analyses UNIVARIÉES

Var. Qualitatives Texte et “Nova Group”

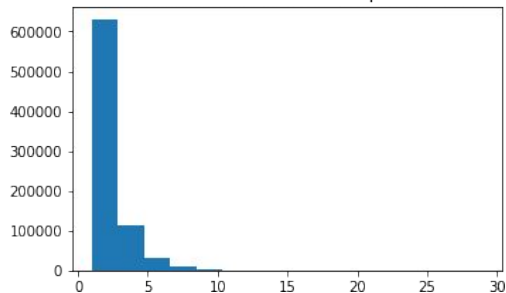
Distribution du nombre de mots pour : countries_en



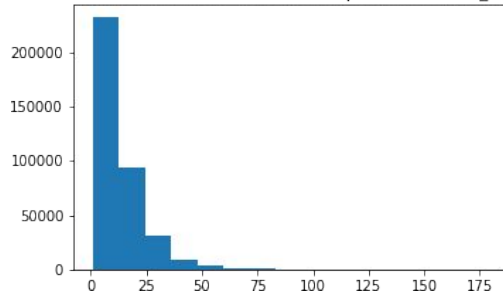
Distribution du nombre de mots pour : product_name



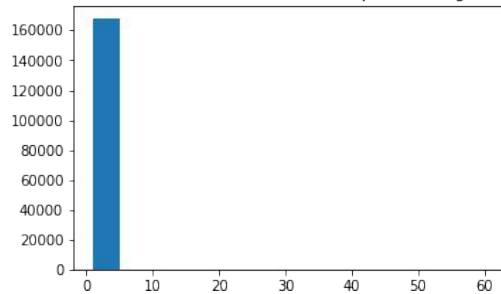
Distribution du nombre de mots pour : brands



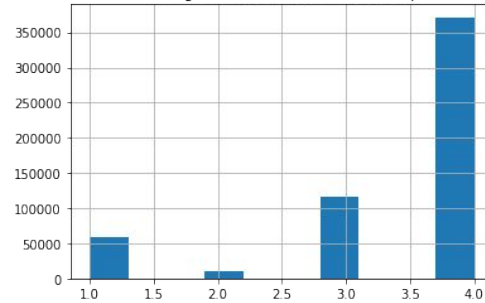
Distribution du nombre de mots pour : additives_en



Distribution du nombre de mots pour : allergens

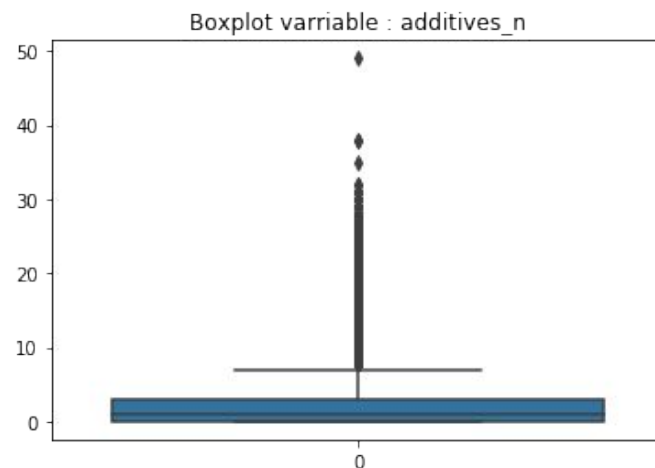


Histogramme variable : Nova Group



Phase 2 : Analyses UNIVARIÉES**Var. Quantitatives Discontinue**

	additives_n	ingredients_from_palm_oil_n	ingredients_that_may_be_from_palm_oil_n
count	640108.000000	640108.000000	640108.000000
mean	2.010467	0.020811	0.069804
std	2.846657	0.144587	0.302400
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	1.000000	0.000000	0.000000
75%	3.000000	0.000000	0.000000
max	49.000000	3.000000	6.000000

**CORRECTION : ÉLIMINATION D'OUTLIERS****additives_n > 30**

Phase 2 : Analyses UNIVARIÉES

Var. Quantitatives

Var. Énergétiques

Var. Nutritionnelles 100g

	energy-kcal_100g	energy_100g
count	1.094588e+06	1.163787e+06
mean	7.944650e+06	5.727473e+36
std	8.309733e+09	6.178739e+39
min	0.000000e+00	0.000000e+00
25%	1.010000e+02	4.180000e+02
50%	2.640000e+02	1.092000e+03
75%	4.000000e+02	1.674000e+03
max	8.693855e+12	6.665559e+42

	fat_100g	saturated-fat_100g	trans-fat_100g	cholesterol_100g	carbohydrates_100g	sugars_100g	fiber_100g
count	1.154305e+06	1.107510e+06	263902.000000	267888.000000	1.153760e+06	1.132704e+06	445640.000000
mean	1.381774e+01	1.227980e+08	0.046625	0.046948	2.850017e+01	1.386366e+01	2.940072
std	1.716359e+02	1.292305e+11	1.062264	1.459132	2.856277e+01	2.008663e+01	5.000099
min	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000e+00	-1.000000e+00	-20.000000
25%	6.000000e-01	1.000000e-01	0.000000	0.000000	3.570000e+00	7.000000e-01	0.000000
50%	6.900000e+00	1.900000e+00	0.000000	0.000000	1.570000e+01	3.900000e+00	1.560000
75%	2.142857e+01	7.220000e+00	0.000000	0.022000	5.357000e+01	2.000000e+01	3.600000
max	1.536795e+05	1.360000e+14	369.000000	300.000000	2.916670e+03	1.350000e+03	439.000000

CORRECTION : ÉLIMINATION D'OUTLIERS

energy_kcal > 1500

val_nutri > 100

energy_100g > 6200

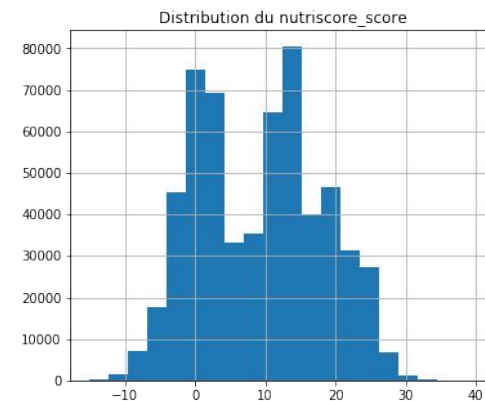
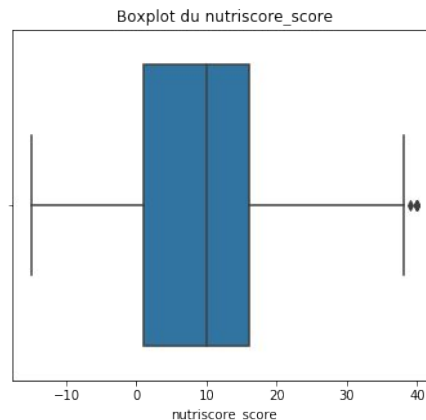
val_nutri < 0

Phase 2 : Analyses UNIVARIÉES

Nutriscore : Score & Grade

N.Score

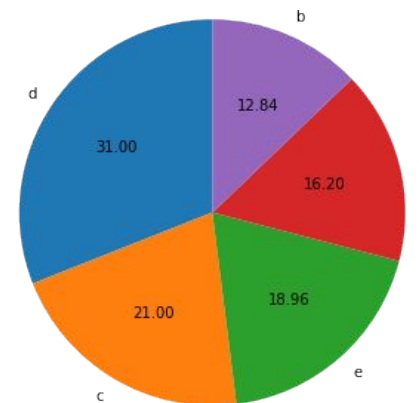
nutriscore_score	
count	582888.000000
mean	9.173872
std	8.914258
min	-15.000000
25%	1.000000
50%	10.000000
75%	16.000000
max	40.000000








N.Grade

```
df.nutriscore_grade.value_counts()
```

```
d    180667
c    122401
e    110512
a     94450
b     74858
Name: nutriscore_grade, dtype: int64
```

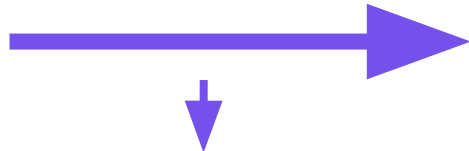


Phase 2 : Analyses UNIVARIÉES**Du Nutriscore_Score au Nutriscore_Grade**

Points		Logo
Solid foods	Beverages	
Min to -1	Waters	
0 - 2	Min - 1	
3 - 10	2 - 5	
11 - 18	6 - 9	
19 - max	10 - max	



APPEL À PROJET



Faisabilité d'un **MOTEUR DE RECOMMANDATION**
(étude exploratoire des données + prototype)

- 1 - Prise de contact / Nettoyage des données
- 2 - Analyses univariées (corrections)
- 3 - Analyses bivariées (corrélations entre var.)
- 4 - Utilisation d'un modèle ML (complétion de data)
- 5 - Réalisation d'un prototype de moteur de rec.
- 6 - Conclusion & Perspectives

Phase 3 : Analyses Bivariées**Retour THÉORIQUE 1/3****Recherche de corrélations linéaires entre 2 variables X et Y**

(et un peu de correction, encore...)

*Est-ce que le fait d'être dans telle ou telle fourchette de valeurs de **X** fait, ou semble faire, qu'on a plus de chance d'être dans telle ou telle fourchette de valeur de **Y** ?*

**Une corrélations entre
2 variables X et Y, ça :**

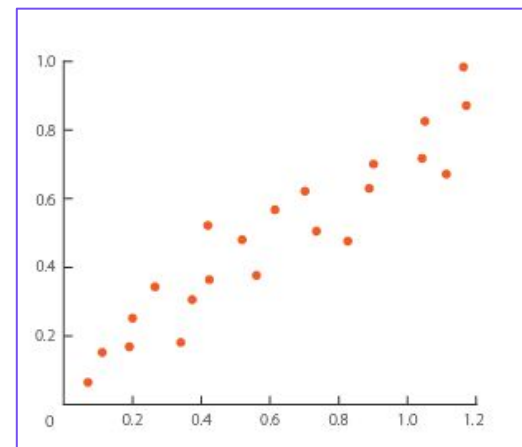
- s'observe **GRAPHIQUEMENT**
- se caractérise **NUMÉRIQUEMENT**
- se valide **STATISTIQUEMENT**

Phase 3 : Analyses Bivariées**Retour THÉORIQUE 2/3****Cas de deux variables QUANTITATIVES**

Graph utilisé : **scatterplot** ou **nuage de points** :

Indicateur numérique : **Coefficient de Pearson**

$$r_{XY} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$



$$-1 < r < 1$$

La fonction **pearsonr** du module **scipy.stats** nous fournira à la fois ce **coefficient** ainsi que la **p-value** du test statistique.

Phase 3 : Analyses Bivariées**Retour THÉORIQUE 3/3****Variable QUANTITATIVE et Variable QUALITATIVE :
ANOVA**

Indicateurs numériques : **Table ANOVA et valeur R2**
obtenus grâce au script **anova.py** et la fonction **ols**
du module **statsmodel.formula.api**

TABLE ANOVA nutriscore_grade en fonction additives_n

	df	sum_sq	mean_sq	F	PR(>F)
Q(vqual)	4.0	2.473285e+05	61832.116508	7554.113932	0.0
Residual	441325.0	3.612344e+06	8.185224	NaN	NaN
SCT		3.859673e+06			
R2	= 0.06408016805367707				

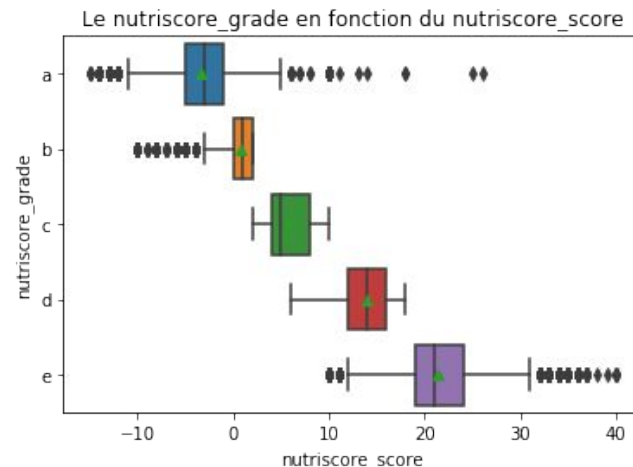
RAPPEL :

$$F = MSB / MSW = [SSB / (C-1)] / [SSW / (n-1)]$$

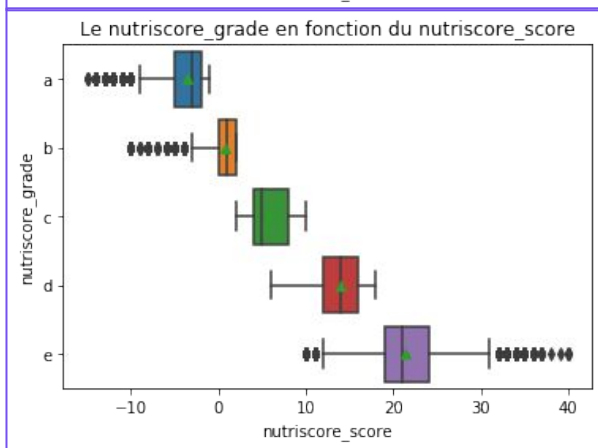
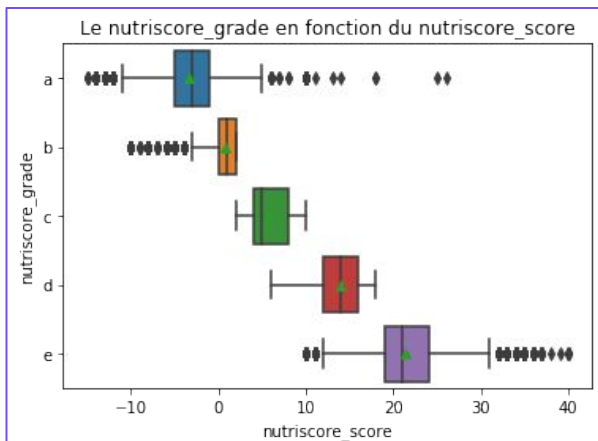
$$R2 = SSB / SST = SSB / (SSB + SSW)$$

SST = Total Sum of Squares**SSB = Sum of Squares Between****SSW = Sum of Squares Within**

$$0 < R2 < 1$$

Graph utilisé : boxplot multiple

Phase 3 : Analyses Bivariées



Nutriscore Score & Grade : correction

```
df[df.nutriscore_grade == "a"]["nutriscore_score"].describe()
```

```
count    94450.000000
mean      -3.396379
std        2.307783
min       -15.000000
25%       -5.000000
50%       -3.000000
75%       -1.000000
max        26.000000
Name: nutriscore_score, dtype: float64
```

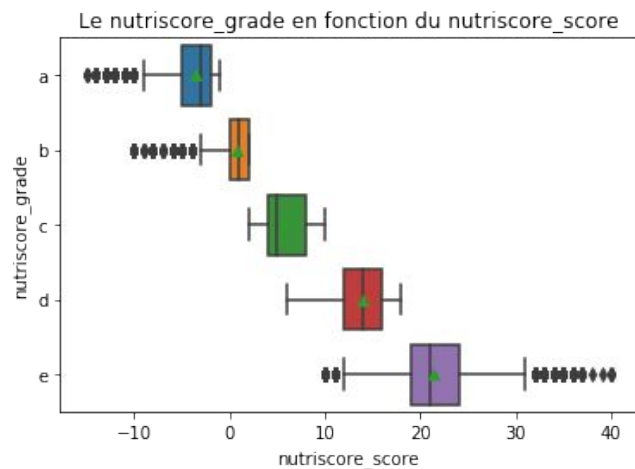
```
len(df[(df.nutriscore_score > -1) & (df.nutriscore_grade == "a")])
```

```
2282
```

```
df.loc[(df["nutriscore_grade"] == "a") & (df["nutriscore_score"] > -1) & (df["nutriscore_score"] <= 2),
       "nutriscore_grade"] = "b"
df.loc[(df["nutriscore_grade"] == "a") & (df["nutriscore_score"] > 2) & (df["nutriscore_score"] <= 10),
       "nutriscore_grade"] = "c"
df.loc[(df["nutriscore_grade"] == "a") & (df["nutriscore_score"] > 10) & (df["nutriscore_score"] <= 18),
       "nutriscore_grade"] = "d"
df.loc[(df["nutriscore_grade"] == "a") & (df["nutriscore_score"] > 18),
       "nutriscore_grade"] = "e"
```


Phase 3 : Analyses Bivariées

Nutriscore Score & Grade : ANOVA



Test Statistique / Table ANOVA, avec le script anova.py

```
anova(df, "nutriscore_grade", "nutriscore_score")
```

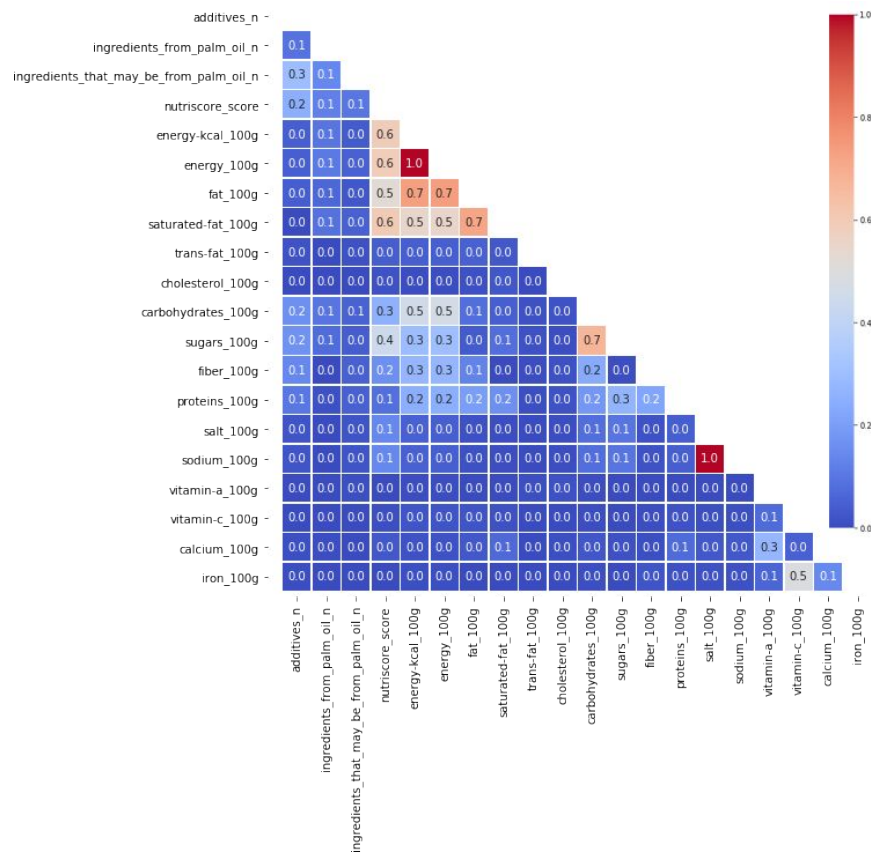
TABLE ANOVA nutriscore_grade en fonction nutriscore_score

	df	sum_sq	mean_sq	F	PR(>F)
Q(vqual)	4.0	4.221583e+07	1.055396e+07	1.499432e+06	0.0
Residual	582883.0	4.102700e+06	7.038635e+00	NaN	NaN
SCT		4.631853e+07			
R2	= 0.9114242011779303				

Phase 3 : Analyses Bivariées

Corrélations entre Var. QUANTITATIVES

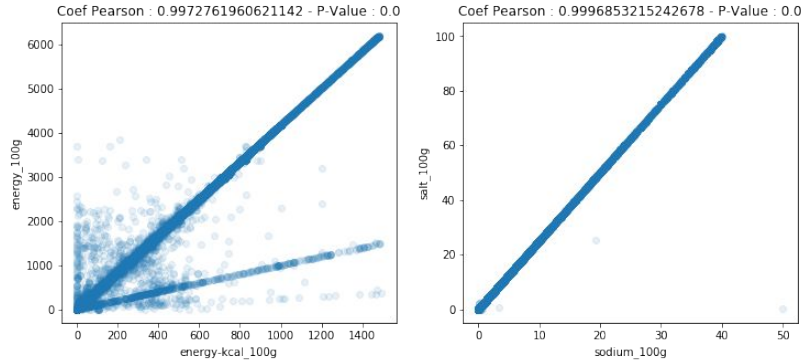
Table de Corrélations



Phase 3 : Analyses Bivariées

Corrélations entre Var. QUANTITATIVES

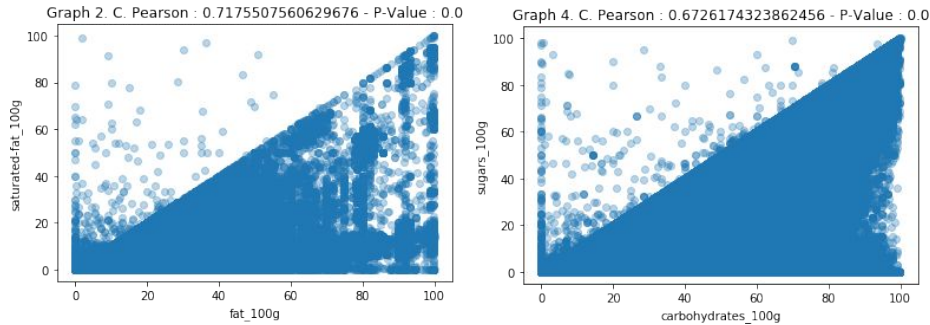
Énergie et Sel



Redondances ! On **élimine** :

- “energy-kcal_100g”
- “sodium_100g”

Graisses et Sucres



Corrections
Création de n.var :

- TSu
- TGsat

Phase 3 : Analyses Bivariées

Var. QUANTITATIVES avec Var. QUALITATIVES - ANOVA

But : Trouver les var. les plus corrélées au **Nutriscore_Grade**

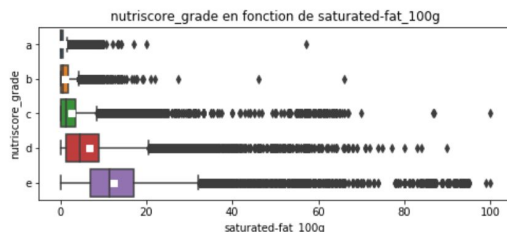


TABLE ANOVA nutriscore_grade en fonction saturated-fat_100g

	df	sum_sq	mean_sq	F	PR(>F)
Q(vqual)	4.0	1.037441e+07	2.593602e+06	65072.039909	0.0
Residual	575531.0	2.293917e+07	3.985739e+01	NaN	NaN
SCT		3.331357e+07			

R2 = 0.31141683776431395

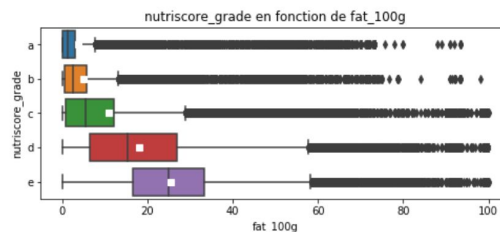


TABLE ANOVA nutriscore_grade en fonction fat_100g

	df	sum_sq	mean_sq	F	PR(>F)
Q(vqual)	4.0	3.399460e+07	8.498651e+06	39831.726509	0.0
Residual	575545.0	1.228005e+08	2.133639e+02	NaN	NaN
SCT		1.567951e+08			

R2 = 0.21680908567687007

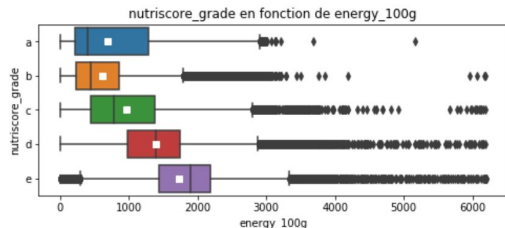


TABLE ANOVA nutriscore_grade en fonction energy_100g

	df	sum_sq	mean_sq	F	PR(>F)
Q(vqual)	4.0	9.231485e+10	2.307871e+10	53129.95376	0.0
Residual	575592.0	2.500270e+11	4.343823e+05	NaN	NaN
SCT		3.423419e+11			

R2 = 0.26965693317154726

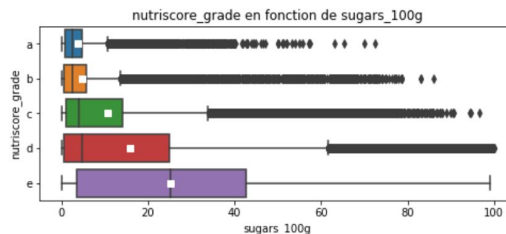


TABLE ANOVA nutriscore_grade en fonction sugars_100g

	df	sum_sq	mean_sq	F	PR(>F)
Q(vqual)	4.0	3.110646e+07	7.776615e+06	28352.255649	0.0
Residual	575534.0	1.578607e+08	2.742856e+02	NaN	NaN
SCT		1.889671e+08			

R2 = 0.16461306505334797

Phase 3 : Analyses Bivariées

Corrélations entre Var. QUALITATIVES

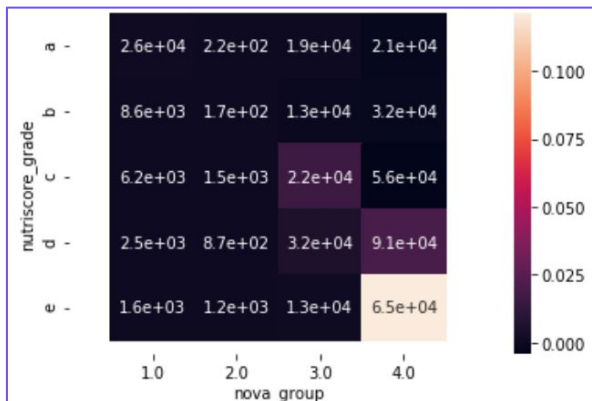
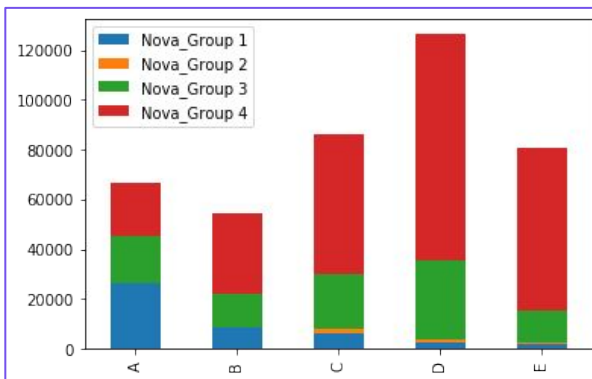
TEST CHI2 entre
nova_group &
nutriscore_grade

Table de
contingence

Soustraction des
“exp. values”

Heatmap

Indicateurs num.
& test stat.



nova_group	1.0	2.0	3.0	4.0	Total
nutriscore_grade					
a	26297	222	18815	21228	66562
b	8583	174	13424	32095	54276
c	6249	1488	22304	55950	85991
d	2544	869	32135	91041	126589
e	1559	1151	12570	65325	80605
Total	45232	3904	99248	265639	414023

```
chi2, p, dof, exp = st.chi2_contingency(cont_tab)
```

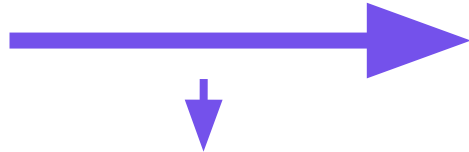
```
87453.48413859907
```

```
0.0
```

```
20
```



APPEL À PROJET



Faisabilité d'un **MOTEUR DE RECOMMANDATION**
(étude exploratoire des données + prototype)

- 1 - Prise de contact / Nettoyage des données
- 2 - Analyses univariées (corrections)
- 3 - Analyses bivariées (corrélations entre var.)
- 4 - Utilisation d'un modèle ML (complétion de data)
- 5 - Réalisation d'un prototype de moteur de rec.
- 6 - Conclusion & Perspectives

Phase 4 : Modèle

Modèle utilisé en vue de compléter la variable nutriscore_grade

Modèle AD-HOC

Modèle “improvisé”. On a gardé le premier avec des résultats “acceptables”

Variables choisies (“X”)

["energy_100g", "fat_100g", "saturated-fat_100g", "TSu", 'proteins_100g', salt_100g',
"sugars_100g"]

Algorithme

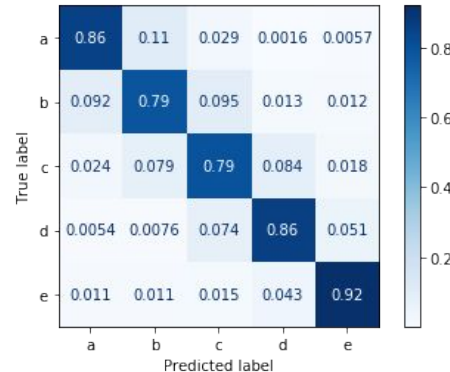
RandomForestClassifier() du module **sklearn**

Résultats

85 % de précision

Retrouvez le processus dans le Notebook :

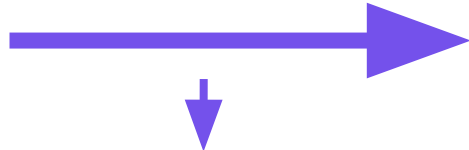
- pré-processing
- modélisation
- prédiction / complétion



On obtient 447000
données
nutriscore_grade



APPEL À PROJET

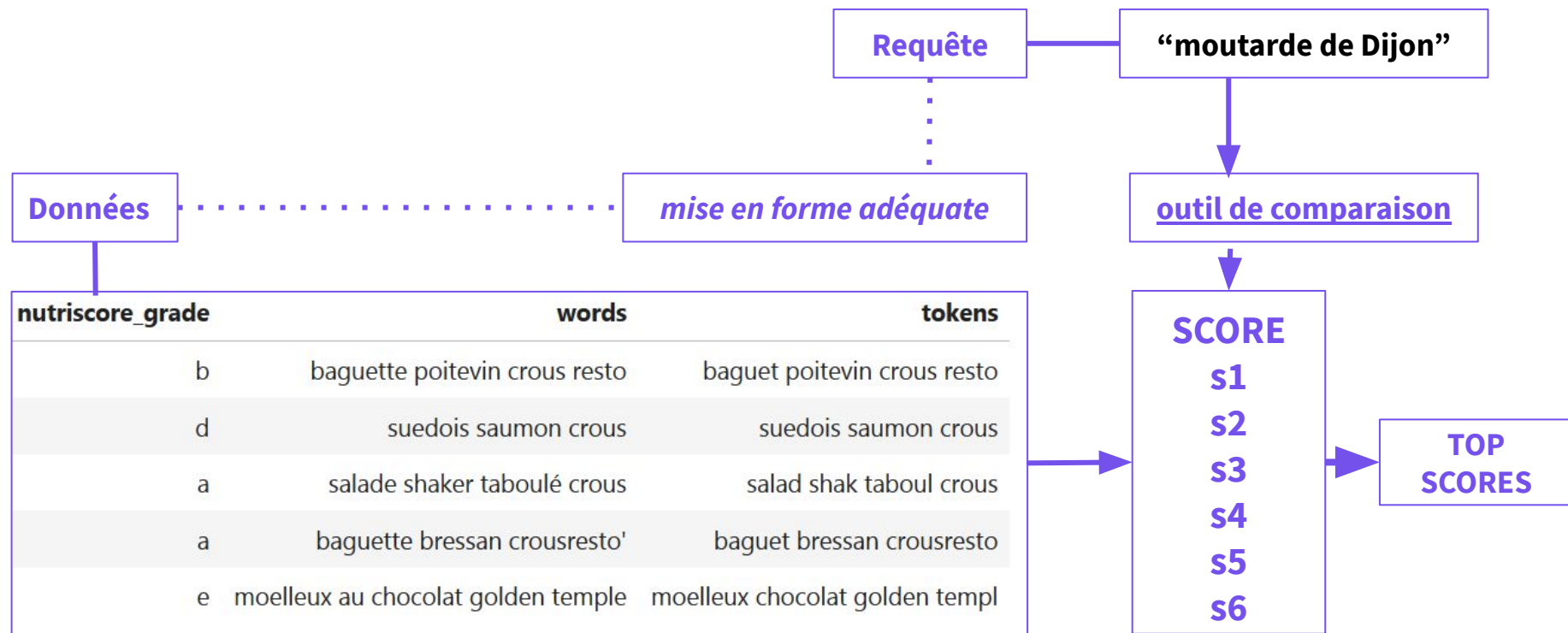


Faisabilité d'un **MOTEUR DE RECOMMANDATION**
(étude exploratoire des données + prototype)

- 1 - Prise de contact / Nettoyage des données
- 2 - Analyses univariées (corrections)
- 3 - Analyses bivariées (corrélations entre var.)
- 4 - Utilisation d'un modèle ML (complétion de data)
- 5 - Réalisation d'un prototype de moteur de rec.
- 6 - Conclusion & Perspectives

Phase 5 : Moteur

Comment cela marche-t-il ?



Phase 5 : Moteur

Traitement des données texte

Passage en
minuscule`.lower()`

Moutarde

moutarde

Elimination des
expressions “parasites”`tokenizer = nltk.RegexpTokenizer('[a-zéèêôûîüâçà]+')`Elimination des
“stopwords”`(nltk) sw = stopwords.words(“french”)`

Désaccentuation

`unidecode.unidecode()`

céréales

cereales

Racinisation :
stemming`(nltk) FrenchStemmer()`

cereales

cereal

Vectorisation

`TfidfVectorizer()`

```
vect = TfidfVectorizer(analyzer = "word")  
matrix = vect.fit_transform(df["tokens"])  
matrix.shape
```

```
(723409, 86762)
```

Phase 5 : Moteur

Prototype du moteur

Requête

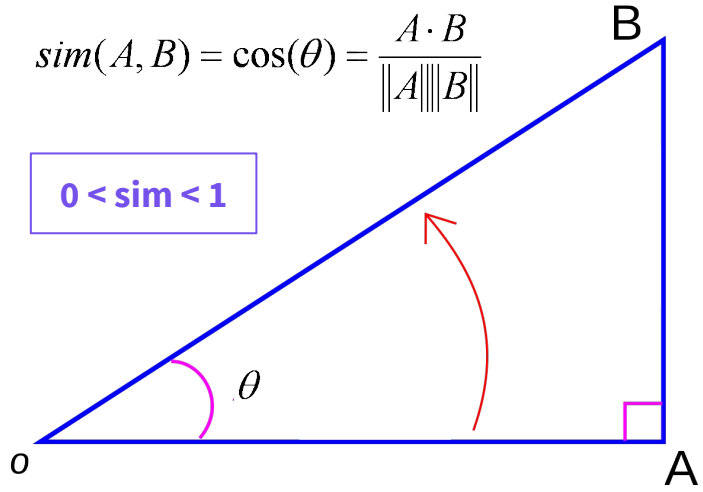
Traitement
sémantique
identique

```
df["score"] = cosine_similarity(query_v, matrix)[0]
```

Moyen de comparaison : **cosine similarity**Tri du dataset en fonction de ce
SCORE

$$\text{sim}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

$$0 < \text{sim} < 1$$



words	score
Jus d'Ananas à base de concentré Carrefour	0.856368
Jus d'ananas à base de concentré Carrefour	0.856368
Jus d'ananas à base de concentré Carrefour	0.856368
Jus d'Ananas Carrefour	0.789972
base de concentré Carrefour,Groupe Carrefour	0.782042
Jus d'Ananas à Base de Concentré	0.764908

Phase 5 : Moteur

Prototype du moteur

Entrez votre recherche : Saumon fumé de Norvège

product_name	nutriscore_grade	nova_group	fiber_100g	score
Saumon fumé Norvège	a	nan	nan	0.882435
Saumon fumé	a	nan	nan	0.687991
Saumon fumé	a	nan	nan	0.687991
Saumon fume	a	nan	nan	0.687991
Pavé de saumon de Norvège	a	nan	nan	0.582578
Saumon atlantique eleve en norvege	a	nan	nan	0.572049
Pavé de Saumon de Norvège	a	1.000000	nan	0.558376
Pavé saumon Norvège 2 × 120 g	a	nan	nan	0.558376
Pavé de saumon Norvège	a	nan	0.000000	0.555499

Phase 5 : Moteur

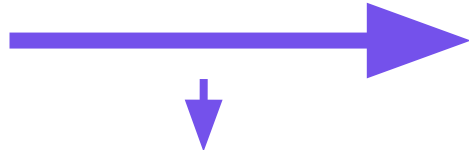
Prototype du moteur

Entrez votre recherche : Caviar

[illegible]



APPEL À PROJET



Faisabilité d'un **MOTEUR DE RECOMMANDATION**
(étude exploratoire des données + prototype)

- 1 - Prise de contact / Nettoyage des données
- 2 - Analyses univariées (corrections)
- 3 - Analyses bivariées (corrélations entre var.)
- 4 - Utilisation d'un modèle ML (complétion de data)
- 5 - Réalisation d'un prototype de moteur de rec.
- 6 - Conclusion & Perspectives

Phase 6 : Conclusion

Avec une approche “data” et des manières
“drastiques”...

On a des résultats !

Alors si en plus...

Conseil et expertise
“métier” dans le
domaine
agro-alimentaire...

Approche plus
réfléchie et meilleure
utilisation des
données et outils déjà
à notre disposition...

Nouveaux outils
ouvrant de nouvelles
perspectives...

Enrichissement des
données, élargissement
du cadre...

Il y a matière à concevoir une application
Open Food Search
riche et attrayante.

