

URCA Reims

RAPPORT DE QUINZAINES

PRÉVISION SUR LE
NUTRI-SCORE



2023

Brunet Alexandre
Ertas Elif

Jupin Manon
Gabet Léo

Kpadondou Carlos
Otogondoua Ememag
Jordhy Jean Jaurès

SOMMAIRE

Introduction	3
1. Premier sprint (du 06/10 au 20/10)	5
1.1. Interface utilisateur	5
1.2. Préparation des données	5
1.3. Algorithme de prévision	6
2. Second sprint (du 21/10 au 27/10)	7
2.1. Préparation des données	7
2.2. Interface utilisateur	7
2.3. Algorithme de prédiction	9
3. Troisième sprint (du 22/10 au 09/11)	10
3.1. Interface utilisateur	10
3.2. Préparation des données	13
3.3. Algorithme de prédiction	14
Conclusion	15
Annexes	16
Table des annexes	17
Annexe 1. Titre de l'annexe	18
Annexe 2. Titre de l'annexe	19
Annexe 3. Titre de l'annexe	20
table des figures	21
Table des tableaux	22
Table des matières	23

INTRODUCTION

Depuis quelques années, l'émergence des enjeux environnementaux s'est accompagnée de la question de la santé et du bien-être physique par l'alimentation. Ces deux questions combinées, en plus de la volonté d'informer le consommateur, a poussé l'Etat à instaurer une norme d'étiquetage des produits transformés : le nutri-score.

Cette norme classe les produits selon leur impact positif ou négatif pour la santé des consommateurs. Le classement en question est illustré par des lettres allant de A (très bon pour la santé) à E (très mauvais pour la santé). La base de ce classement repose sur les composants nutritionnels des produits. Notre objectif est de déterminer quels sont les éléments nutritionnels qui influent sur le classement des produits et dans quelle mesure.

Pour répondre à cette question, nous nous sommes appuyés sur une base de données regroupant un peu plus de 52 260 produits ayant chacun leurs caractéristiques nutritionnelles (variables explicatives) et leur nutri-score variable cible (tableau 1).

<i>Variables</i>	Description
<i>energy_100g</i>	Quantité de kilocalories pour 100g
<i>fat_100g</i>	Quantité de matières grasses pour 100g
<i>saturated-fat_100g</i>	Quantité de graisses saturées pour 100g
<i>trans-fat_100g</i>	Quantité de graisses transformées pour 100g
<i>cholesterol_100g</i>	Quantité de cholestérol pour 100g
<i>carbohydrates_100g</i>	Quantité de carbohydrates pour 100g
<i>sugars_100g</i>	Quantité de sucres pour 100g
<i>fiber_100g</i>	Quantité de fibres pour 100g
<i>proteins_100g</i>	Quantité de protéines pour 100g
<i>salt_100g</i>	Quantité de sel pour 100g
<i>sodium_100g</i>	Quantité de sodium pour 100g
<i>vitamin-a_100g</i>	Quantité de vitamine A pour 100g
<i>vitamin-c_100g</i>	Quantité de vitamine C pour 100g
<i>calcium_100g</i>	Quantité de calcium pour 100g
<i>iron_100g</i>	Quantité de fer pour 100g
<i>nutrition_grade_fr</i>	Notre variable cible qui est décrite par des lettres allant de A à E exprimant la qualité nutritionnelle du produit.

Tableau 1 : Description des variables utilisées.

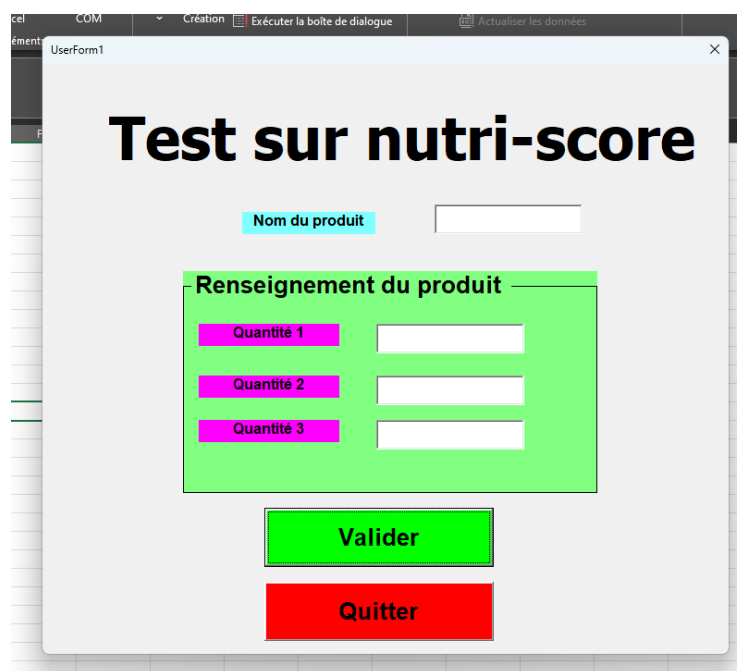
Cette base est présente sur le site de l'open-data du gouvernement¹. Grâce à cela, nous allons mobiliser des outils d'apprentissage statistique pour qu'un consommateur, en entrant les informations du produit, puisse évaluer de la pertinence du nutri-score d'un produit (ou alors connaître le son nutri-score dans le cas où il n'y a pas accès).

¹ <https://www.data.gouv.fr/fr/datasets/open-food-facts-produits-alimentaires-ingredients-nutrition-labels/>

1. Premier sprint (du 06/10 au 20/10)

1.1. Interface utilisateur

Pour notre première interface, nous mis en place d'un formulaire avec 4 entrées composés de textBox et de label (figure 1). Ces éléments permettent d'entrer le nom du produit et 3 valeurs au choix, avec bouton de validation et fermeture du formulaire. De plus, le but de cette première interface est de préparer le terrain pour la suite. En effet, on ne sait pas combien de variables, nous allons décider de conserver.



The image shows a screenshot of a Windows application window titled "Test sur nutri-score". The window has a standard Windows title bar with buttons for "Création", "Exécuter la boîte de dialogue", and "Actualiser les données". The main content area is light gray and contains the following elements:

- A large black title "Test sur nutri-score".
- A label "Nom du produit" in a light blue box, followed by a white text input field.
- A green-bordered box titled "Renseignement du produit" containing three labels "Quantité 1", "Quantité 2", and "Quantité 3" in pink boxes, each followed by a white text input field.
- Two buttons at the bottom: a green "Valider" button and a red "Quitter" button.

Figure 1 : Première version de l'interface d'entrée.

1.2. Préparation des données

Pour la préparation des données, en plus des variables que nous avons énumérées, il existe d'autres variables dans notre base de données. Cependant, celles-ci présentent soit des caractéristiques inutiles, soit une complexité qui les rend difficilement exploitables, ou encore une faible pertinence pour un utilisateur lambda. En conséquence, nous avons pris la décision de les exclure de notre analyse. De plus, notre choix de focaliser notre attention sur les valeurs nutritionnelles renforce notre justification pour leur suppression.

1.3. Algorithme de prévision

Pour le choix de l'algorithme de prévision, nous hésitons entre plusieurs modèles : la régression linéaire et la régression logistique multi-groupe ordinale. Pour le premier, nous le connaissons bien et savons comment l'utiliser. De plus, nous avons pensé qu'il nous suffirait de convertir le Nutri-Score en chiffres (au lieu de A, B, C, D et E, nous aurions 1, 2, 3, 4 et 5). Ainsi, lorsque l'utilisateur renseigne les informations, la réponse sera comprise entre 1 et 5, que nous convertirons en lettres. Par exemple, si la réponse est 3, nous renverrons à l'utilisateur que le Nutri-Score est C. Dans le cas où la réponse de l'algorithme renvoie un chiffre non entier, nous avons pensé à arrondir pour conclure.

D'un autre côté, la régression logistique multi-groupe ordinale nous semble la plus appropriée pour notre objectif. En effet, il s'agit de groupes distincts plutôt que de simples valeurs quantitatives, et chaque groupe transmet un message spécifique à l'utilisateur. De plus, ces différents groupes suivent une logique ordinale (A est meilleur que B, B est meilleur que C, etc.), ce qui nous pousse à privilégier ce modèle plutôt que la régression linéaire. Ainsi, la réponse de l'algorithme fournira à l'utilisateur une probabilité associée à chaque groupe que nous visualiserons pour obtenir une vue d'ensemble de chaque groupe.

2. Second sprint (du 21/10 au 27/10)

2.1. Préparation des données

L'objectif de cette seconde quinzaine c'est de préciser notre algorithme de prédiction. Plus clairement, nous voulons à partir de toutes les variables explicatives que nous avons à disposition, en réduire leur nombre pour conserver celles qui sont vraiment utiles.

Pour atteindre cet objectif nous avons décidé d'utiliser les algorithmes AIC et BIC pour la sélection des données. Ces algorithmes nous ont donné chacun leur modèle optimal (celui qui réduit au maximum l'erreur de classification). Ensuite, nous avons comparé ainsi leur erreur de classification respectif pour ainsi choisir lequel entre le modèle optimal au sens de l'AIC et le modèle optimal au sens du BIC est le meilleur (celui qui minimise l'erreur de classification).

Il faut savoir que ces algorithmes ont certes un objectif commun (réduire au mieux l'erreur de classification), mais pas de la même façon. En effet l'AIC conserve au mieux l'information contenue dans les variables. Le BIC pénalise les modèles complexes en étant plus sévère dans sa sélection : il privilégie les modèles simples (avec le moins de variables possibles). Finalement, ces deux algorithmes ont sélectionné exactement les mêmes variables qui apparaissent dans le formulaire (figure 2).

Enfin, nous avons établi le lien entre VBA et Python. Plus précisément, quand on lance le calcul, la macro lance le fichier py qui fait tourner le modèle. Puis en fonction des informations entrées, un message apparaît en disant le chiffre associé au nutri-score (de 0 à 4). Par exemple si le nutri-score est A, le message est « Le Nutri-score est 0 ».

2.2. Interface utilisateur

2.2.1. Interface d'entrée

La sélection des variables est importante car elle permet de réduire le nombre d'informations que l'utilisateur entre dans le formulaire. En effet, moins il doit entrer d'informations, plus il aura tendance à l'utiliser fréquemment et de manière plus ludique. Ces variables sélectionnées seront justement les informations que l'utilisateur devra rentrer dans le formulaire (figure 2).

On reprend ici le principe de TextBox et label qu'on renomme selon le nom des variables, par exemple la variable énergie aura le label : label_energie et la textBox suivante : TextBox_energie. L'intérêt est de comprendre l'utilisation de ces variables dans le code VBA.

Ensuite lors du traitement, on range les valeurs des textBox à la fois dans un historique et en même dans des variables types numériques qui seront envoyés sur Python. Après sur python, on recevra donc une réponse, objectif prévu pour le sprint prochain.

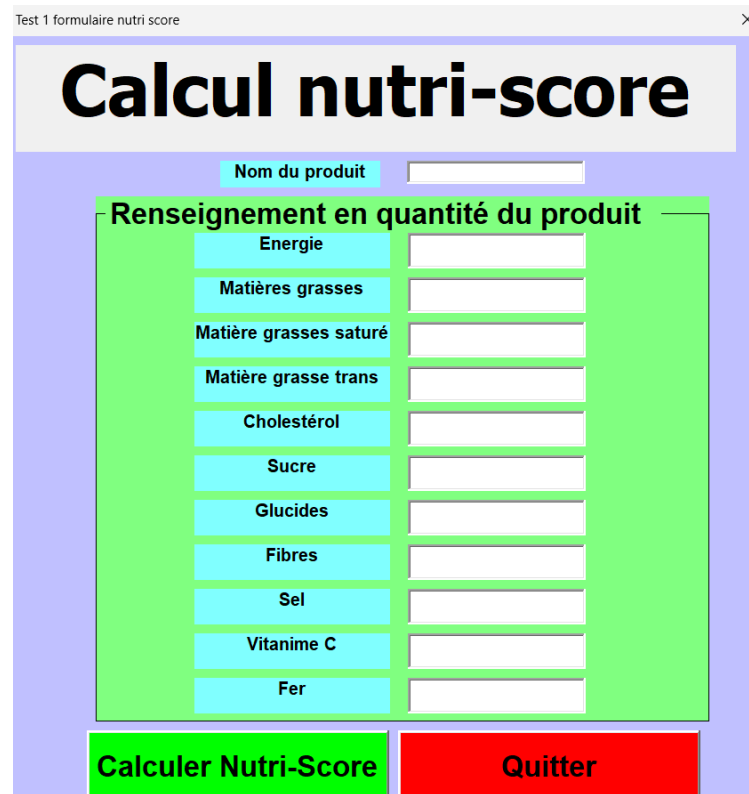


Figure 2 : Seconde version de l'interface d'entrée avec les variables du premier modèle optimal.

2.2.2. Interface de sortie

Afin d'informer au mieux les consommateurs qui utiliseront notre application, il est essentiel de réaliser un Dashboard final. C'est une étape importante puisque ce Dashboard est ce qui permet de rendre visible nos résultats auprès des clients. Nous avons alors débuté la réflexion sur les informations que nous voulions voir apparaître après calcul du nutri-score. La chose la plus évidente était de faire apparaître de manière simple et précise le résultat de notre algorithme de prévision. Pour cela, nous avons décidé de réaliser un graphique de type « jauge » (figure 3). Cette représentation est adaptée puisqu'elle permet une visualisation instantanée du résultat et est facile de compréhension.

Pour créer ce graphique dans Excel, nous avons fusionné deux types de graphiques. Tout d'abord, nous avons utilisé un graphique de type "Anneau" pour représenter les différentes catégories. Chaque couleur de l'anneau est associée à une lettre, par exemple, le vert foncé correspond à la lettre A, et le rouge à la lettre E. Ensuite, pour réaliser le curseur, nous avons

créé un graphique circulaire qui affiche seulement une petite portion du cercle. En fonction de la lettre obtenue, nous avons ajusté les valeurs du graphique circulaire pour déplacer la partie visible du cercle vers la couleur associée au résultat.



Figure 3 : Première interface de sortie.

2.3. Algorithme de prédiction

Nous allons utiliser ce que les algorithmes de sélection nous ont retourné. Autrement dit, nous allons effectuer une première régression logistique ordinaire avec le meilleur modèle que nous avons trouvé avec les différents algorithmes de sélection de variables.

3. Troisième sprint (du 22/10 au 09/11)

3.1. Interface utilisateur

3.1.1. Interface d'entrée

L'idée reste basée sur le principe de la version précédente pour les variables en VBA. Nous avons également commencé à rajouter une sécurité pour vérifier la valeur d'une entrée. En effet, Python n'accepte que le format "0.0" et non "0,0". Pour l'instant, seule la variable énergie est vérifiée donc si on rentre une valeur incorrecte, alors un message nous alerte et nous devons ressaisir une nouvelle valeur sous le bon format. Cette idée sera appliquée sur l'ensemble des variables quand nous serons au sprint 5. La structure du code VBA a été revue pour présenter chaque étape du formulaire, une première partie sur la mise en place du calcul du nutri-score avec le lien python (envoi & réponse), et une seconde partie sur les mises en forme du circulaire (sécurité concernant la nature de la valeur entrée par exemple).

Maintenant, voyons comment se présente l'interface d'entrée. D'abord, lorsque nous arrivons sur la feuille Excel, nous voyons la jauge du dashboard ainsi qu'un bouton « Mon Historique » (figure 4). Lorsque l'on clique sur la jauge, le formulaire apparaît pour que l'on puisse entrer les informations (figure 5). Après avoir correctement entrer les informations (pour l'énergie seulement pour l'instant) puis valider, un récapitulatif des informations s'affiche (figure 6) avant de lancer un algorithme et de mettre ces informations dans une feuille dédiée à l'historique (figure 7). Il faut noter que nous pouvons naviguer à travers les feuilles du fichier qu'avec les boutons dédiés. Enfin, après avoir saisis les informations, l'algorithme tourne pendant quelques secondes puis on est redirigé vers le dashboard (figure 3).

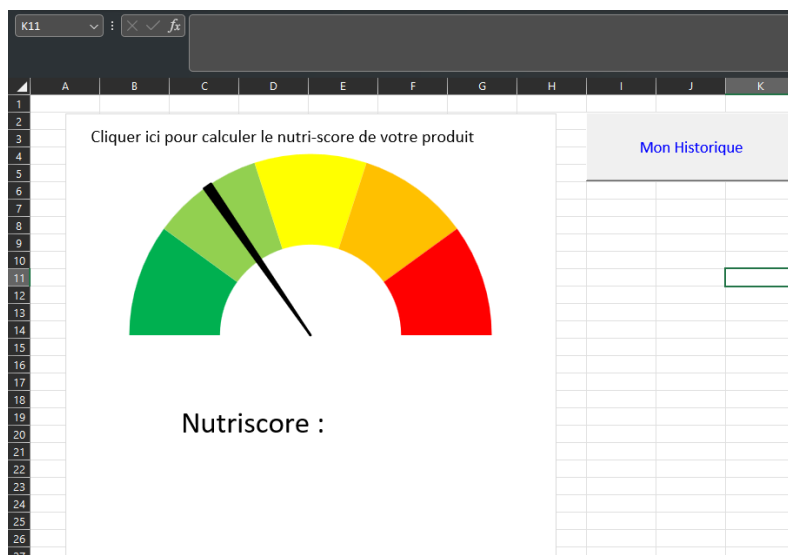


Figure 4 : Présentation de la feuille Excel après ouverture du fichier.

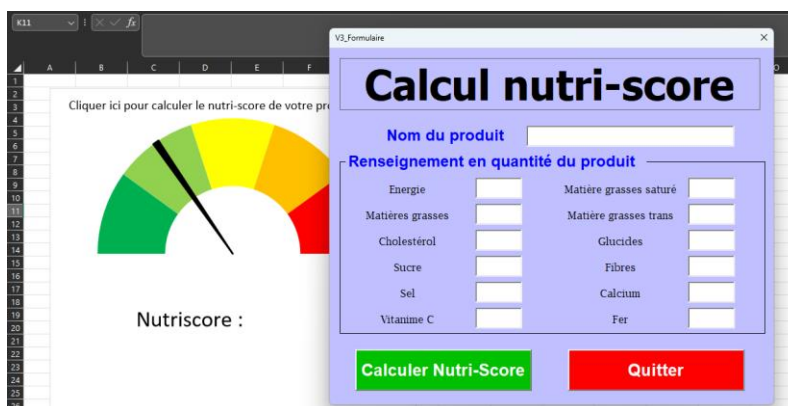


Figure 5 : Apparition du formulaire lorsque l'on clique sur la jauge.

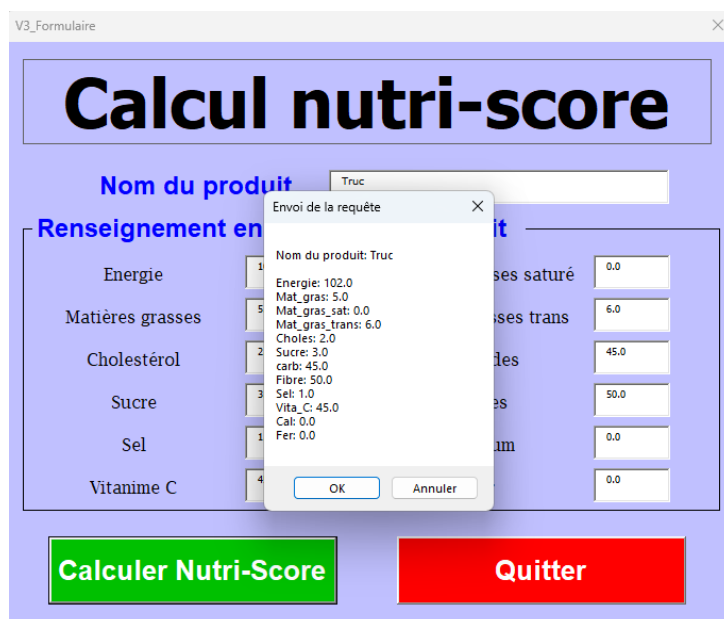


Figure 6 : Récapitulatif des informations saisies.

Nom du produit	Energie	Matières grasses	Matières grasses saturé	Matières grasses saturé trans	Cholestérol	Sucre	Carb	Fibres	Sel	Vitamine c	Cal	Fer	Résultat Nutri-score
Opened													
qsgfd	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	D
test1	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	D
test	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	D
Fromage	1	1	6	7	2	3	8	9	4	5	10	11	E
no,	0	1											
nonnn	1	1	2	2	1	1	2	2	1	1	2	2	E
Truc	102	5	0	6	2	3	45	50	1	45	0	0	E

Retour

Figure 7 : Historique des informations.

3.1.2. Interface de sortie

Pour notre dashboard, on conserve toujours l'idée de la jauge, qui est pour le moment la seule partie de notre dashboard final qui est conceptualisée (ou créée). Cependant, pour compléter ce dashboard, nous avons réfléchi et pré-conceptualiser d'autres informations qui apparaîtront. Finalement, voici les éléments qui composeront le dashboard final (figure 8) :

- Nutri-score : Graphique « jauge » réalisé pour le sprint 2. Le résultat est accompagné d'un commentaire encourageant le consommateur à continuer si on a un bon nutri-score. Si au contraire on obtient un mauvais score, le commentaire invitera le consommateur à consommer avec modération.
- Classes : Graphique circulaire affichant les pourcentages d'appartenance aux classes « A, B, C, D, E ». Un tel graphique a pour but de mettre en avant si le résultat obtenue est nette ou non. Dans l'exemple ci-dessus, il y a 80% de chance que le résultat soit A. Nous sommes donc quasiment persuadés du résultat que nous avançons. En revanche, il se peut que nous ayons des résultats serrés comme 53% pour A contre 47% pour B. Le résultat affiché sera A. De plus, lorsque nous aurons estimé l'erreur théorique du modèle utilisé, nous en informerons le client. Par exemple : « Il y a 30% de chance que les probabilités affichées soient fausses. »
- Classement : Top 3 des nutriments expliquant le plus le nutri-score. Ce classement permet de savoir quels nutriments influencent le plus le résultat. Ainsi, le consommateur sait ce qu'il doit limiter ou non pour améliorer son score. Le classement affiché dans l'exemple n'est pas réel, nous n'avons pas encore cherché ce résultat.
- Sucre/Sel : Donne l'équivalent du taux de sucre et de sel en une unité visuelle. Dans cette exemple nous avons sucre et sel car ce sont les deux premiers de notre classement précédent. Nous verrons à changer les nutriments mis en avant selon notre classement réelle.

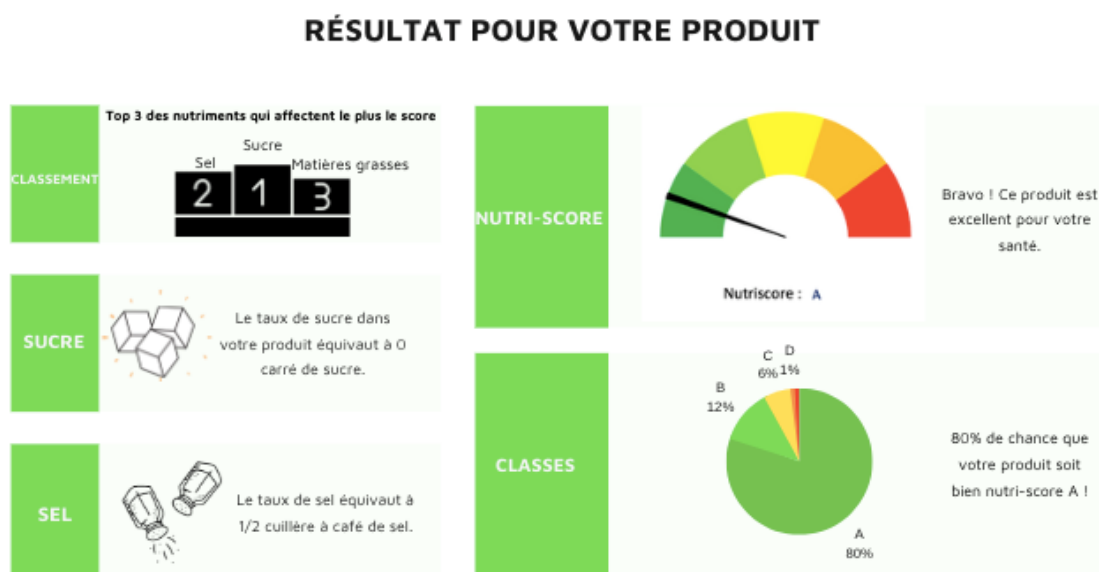


Figure 8 : Maquette de dashboard final.

3.2. Préparation des données

3.2.1. Préparation du modèle prédictif

Nous avons poursuivi la création de modèle pour ensuite les comparer. Dans cette optique, nous avons entraîné 3 modèles supplémentaires : Lasso, Ridge puis le modèle initial (régression logistique ordinale brute, sans sélection et sans pénalisation) que nous avons oublié d'entraîner dès le début. Nous avons ainsi comparé tous les modèles que nous avons entraînés avec leur erreur de classification (tableau 2).

Modèle	Erreur de classification (en %)
<i>Initial (global)</i>	26,1
<i>Optimal au sens de l'AIC</i>	28
<i>Optimal au sens du BIC</i>	28
<i>Optimal Lasso</i>	27,5
<i>Optimal Ridge</i>	43,6

Tableau 2 : Comparaison des modèles utilisés.

En les comparant tous ensemble, nous avons décidé de choisir le modèle qui minimise l'erreur de classification et c'est le modèle global initial qui semble être le meilleur candidat.

3.2.2. Etablissement du lien VBA-Python

Nous souhaitons intégrer VBA (Visual Basic for Applications) et Python pour automatiser la prédiction du Nutri-score à partir de données saisies par les utilisateurs dans un formulaire Excel. L'objectif est de fournir une solution permettant d'évaluer et d'afficher le Nutri-score en temps réel en se basant sur un modèle de prédiction Python. La relation python vba s'effectue selon le processus suivant :

1. Les utilisateurs saisissent des données dans un formulaire Excel.
2. Les données saisies sont stockées dans des variables VBA pour une manipulation ultérieure.
3. Ces variables VBA sont utilisées pour créer une structure de données JSON (format {« clé »: valeur}).
4. Une API Python a été développée pour réaliser la prédiction du Nutri-score. Cette API est hébergée à l'adresse <http://kpcarlos.pythonanywhere.com/test/>

Le script de l'API décharge les deux modèles python contenus dans le pickle, puis récupère les données issus de la requête reçue. Il effectue ensuite la prédiction et recode le label correspondant à la valeur prédite (changer les chiffres en lettres). Enfin le label est retourné par l'API.

5. Les données au format JSON sont envoyées à cette API Python pour effectuer la prédiction.
6. L'API Python renvoie une réponse au format JSON contenant la valeur prédite du Nutri-score.
7. Une fonction appelée "jsonconverter" est utilisée pour interpréter le format JSON retourné et extraire la valeur du Nutri-score prédit.

3.3. Algorithme de prédiction

Nous avons entraîné le modèle que nous avons gardé puis nous l'avons enregistré dans un format pickle pour une utilisation ultérieure.

En outre, pour avoir une première vue sur le graphique qui retourne les différentes probabilités d'appartenance aux groupes (ou classes), nous avons utilisé 9 échantillons et nous les avons évalués avec le modèle. Avec les sorties du modèles, nous avons créé 9 graphiques circulaires.

CONCLUSION

Tapez votre conclusion.

ANNEXES

TABLE DES ANNEXES

Annexe 1. Titre de l'annexe.....	18
Annexe 2. Titre de l'annexe.....	19
Annexe 3. Titre de l'annexe.....	20

Annexe 1. Titre de l'annexe

Annexe 2. Titre de l'annexe

Annexe 3. Titre de l'annexe

TABLE DES FIGURES

Figure 1 : Première version de l'interface d'entrée.....	5
Figure 2 : Seconde version de l'interface d'entrée avec les variables du premier modèle optimal.....	8
Figure 3 : Première interface de sortie.	9
Figure 4 : Présentation de la feuille Excel après ouverture du fichier.	11
Figure 5 : Apparition du formulaire lorsque l'on clique sur la jauge.....	11
Figure 6 : Récapitulatif des informations saisies.	11
Figure 7 : Historique des informations.....	12
Figure 8 : Maquette de dashboard final.....	13

TABLE DES TABLEAUX

Tableau 1 : Description des variables utilisées.	3
Tableau 2 : Comparaison des modèles utilisés.	13

TABLE DES MATIERES

Introduction	3
1. Premier sprint (du 06/10 au 20/10)	5
1.1. Interface utilisateur	5
1.2. Préparation des données	5
1.3. Algorithme de prévision	6
2. Second sprint (du 21/10 au 27/10)	7
2.1. Préparation des données	7
2.2. Interface utilisateur	7
2.2.1. Interface d'entrée	7
2.2.2. Interface de sortie	8
2.3. Algorithme de prédiction	9
3. Troisième sprint (du 22/10 au 09/11)	10
3.1. Interface utilisateur	10
3.1.1. Interface d'entrée	10
3.1.2. Interface de sortie	12
3.2. Préparation des données	13
3.2.1. Préparation du modèle prédictif	13
3.2.2. Etablissement du lien VBA-Python	14
3.3. Algorithme de prédiction	14
Conclusion	15
Annexes	16
Table des annexes	17
Annexe 1. Titre de l'annexe	18
Annexe 2. Titre de l'annexe	19

Annexe 3. Titre de l'annexe	20
table des figures.....	21
Table des tableaux.....	22
Table des matières.....	23