Nutrition Based Analysis Using Machine Learning Model

Sahitya Singh
School of Computer Science and
Artificial Intelligence (SCAI)
Vellore Institute of Technology
Bhopal, India
sahitya.23bai10570@vitbhopal.ac.in

Shikhar Srivastava
School of Computer Science and
Artificial Intelligence (SCAI)
Vellore Institute of Technology
Bhopal, India
shikhar.23bai10613@vitbhopal.ac.in

Harshit Yadav
School of Computer Science and
Artificial Intelligence (SCAI)
Vellore Institute of Technology
Bhopal, India
harshit.23bai10556@vitbhopal.ac.in

Naman Gupta
School of Computer Science and
Artificial Intelligence (SCAI)
Vellore Institute of Technology
Bhopal, India
naman.23bai10625@vitbhopal.ac.in

Aarya Nema
School of Computer Science and
Artificial Intelligence (SCAI)
Vellore Institute of Technology
Bhopal, India
aarya.23bai10786@vitbhopal.ac.in

Anshu Kumar
School of Computer Science and
Artificial Intelligence (SCAI)
Vellore Institute of Technology
Bhopal, India
anshu.23bai10519@vitbhopal.ac.in

I. Abstract

The increasing incidence of non-communicable diseases (NCDs) due to inappropriate dietary practices has fueled the necessity for sophisticated instruments in nutritional analysis and intervention. This paper proposes a machine learning-based method to assess the nutritional value of packaged food items. Based on data from OpenFoodFacts, we created and compared various classification models—Random Forest, Logistic Regression, and different Support Vector Machine (SVM) models—to classify health ratings (A to E) according to ingredient makeup and nutrition labels. Of the models tested, the Random Forest classifier had the best accuracy of around 89%, showing its effectiveness in dealing with class imbalance and intricate nutritional data. The system developed from this is an enabling tool for individualized nutrition counseling and consumer education. Directions for the future involve incorporating this model into mobile systems to enable real-time analysis of foods and improved decision-making.

II. Introduction

Nutrition science explores the intricate relationship between dietary intake, health outcomes, and disease prevention [1].Nutrition is fundamentally linked to both physiological and biochemical functions, as it explains how food components supply energy or contribute to the formation of body tissues [2]. The processes are vital to life and affect overall health, physical development, and the prevention of disease.These mechanisms are vital for life and significantly influence overall well-being, physical development, and the prevention of illnesses. Nutrition plays a crucial role by clarifying how diet impacts health and quality of life. As noted by Melaku et al. [3], most diseases are not inherited but are primarily the result of poor dietary patterns, leading to substantial global healthcare expenditures. Nutritional science provides crucial insights into the root causes of diseases and enhances our ability to develop effective preventive measures. It also plays a vital role in examining how diseases are distributed across populations. Popkin et al. [4] report significant shifts in global dietary habits, which are closely linked to changes in disease patterns. These ongoing developments highlight the essential role of nutrition in tackling widespread health issues such as obesity, diabetes, and cardiovascular diseases [5].

Beyond its role in preventing disease, nutrition is also essential in managing and treating a wide range of health conditions [6]. As highlighted in a 2020 report by the American Diabetes Association, medical nutrition therapy—an integral aspect of managing diseases such as diabetes—is grounded in well-established nutritional science principles [7]. Furthermore, the field of nutrition is advancing swiftly with the integration of genomics and the emergence of personalized nutrition. This innovative approach customizes dietary guidance according to an individual's genetic makeup [8], holding great promise for transforming disease prevention strategies and improving overall health outcomes.

While traditional methods rely heavily on clinical and observational studies, modern advancements in artificial intelligence (AI), particularly machine learning (ML), offer transformative capabilities. These technologies enable the processing of vast nutritional datasets, revealing hidden patterns and supporting data-driven dietary recommendations [9, 10]. Personalization of diet to predictive prevention models of disease, the potential uses of AI in nutrition are diverse and far-reaching. Integration of AI applications in nutrition introduces technological advancement that shifts the paradigm of dietary interventions [11]. AI approaches have much potential in this era of data to transform the way we understand, track, and optimize nutritional outcomes.

A. *Background*

Traditional applications like MyFitnessPal and Yuka use barcode scanning for nutritional display but lack personalized health assessments. Nutritional databases such as USDA and OpenFoodFacts provide valuable data but are static and do not offer health impact evaluations.

While barcode scanners offer ease of use, they are often limited by the quality of their data sources. Machine learning has entered this domain with applications that seek to classify foods as healthy or unhealthy using ingredients and nutrition labels. However, many systems lack transparency, personalization, and adaptability.

NutriScore differentiates itself by combining multiple methodologies: it blends data retrieval, real-time scanning, user customization, and robust classification models, offering a more holistic view of nutrition.

1. *Details*

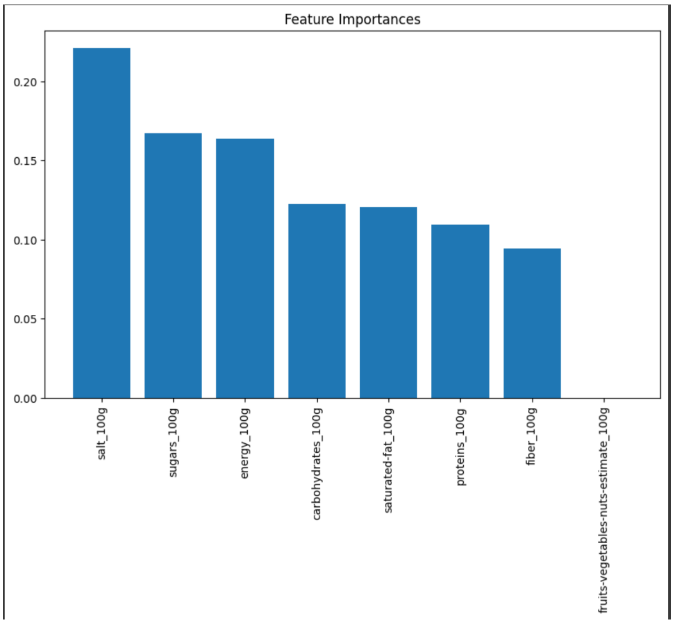


Figure I: Feature Importance

Here we create a hybrid model using four various machine learning algorithms, namely logistic regression, decision tree classifier, random forest classifier, SVM (Scalar Vector Machine). These will than analyze and sort the food product based on the ingredients provided by the company itself and these are further broken into nutritional value for each ingredient 100g of product. It will then classify the products it scanned and give them a health label from A to E, A being the healthiest and E being the unhealthiest. We Created similar models with the same data to see which one performs better in the packaged food classification based on various parameters

III. Frameworks

The study was started by the installation of a few Python models/Libraries, as the Machine Learning model is based on Python. These models are are follows-

A. *Matplotlib [11]*

Matplotlib is a comprehensive, free, and open-source plotting library for the Python programming language, designed to work well with NumPy and the broader SciPy stack. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits, as well as a procedural interface (pyplot) for creating various static, animated, and interactive visualizations quickly. It features extensive capabilities for generating numerous plot types including line plots, scatter plots, bar charts, histograms, error charts, contour plots, and 3D surfaces.

B. *Scikit-Learn(sklearn) [12]*

scikit-learn (formerly scikits.learn and also known as sklearn) is a free and open-source machine learning library for the Python programming language. It features various classification, regression, and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means, and DBSCAN.

C. *Numpy [13]*

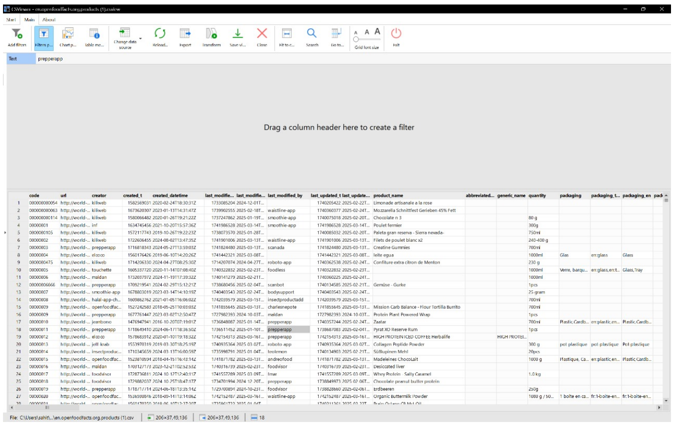
NumPy brings the computational power of languages like C and Fortran to Python, a language much easier to learn and use. With this power comes simplicity: a solution in NumPy is often clear and elegant.

D. *Pandas [14]*

Pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language.

IV. Methods

A dataset from opensource websites in the csv format was further refined. This database included the name of the packaged food products, the barcode of the products, the nutrition score ranging from A to E (A being the best and E being the worst) ,the list of the ingredients of the packaged food products and lastly nutrition facts of product per 100g. Another dataset listing critical intake values of nutrients such as carbs and fats for various medical conditions.



	code	url	creator	created_t	created_datetime	last_modifie...	last_modifie...	last_modified_by	last_updated_t	last_update...	product_name	abbreviated...	generic_name	quantity	packaging	packaging_t...	packaging_en	pack
1	000000000054	http://world-...	kiliweb	1582569031	2020-02-24T18:30:31Z	1733085204	2024-12-01T...		1740205422	2025-02-22T...	Limonade artisanale a la rose							
2	000000000063	http://world-...	kiliweb	1673620307	2023-01-13T14:31:47Z	1739902555	2025-02-18T...	waistline-app	1740360377	2025-02-24T...	Mozzarella Schnittfest Gerieben 45% Fett							
3	000000000114	http://world-...	kiliweb	1580066482	2020-01-26T19:21:22Z	1737247862	2025-01-19T...	smoothie-app	1740075018	2025-02-20T...	Chocolate n 3			80 g				
4	00000001	http://world-...	inf	1634745456	2021-10-20T15:57:36Z	1741986528	2025-03-14T...	smoothie-app	1741986528	2025-03-14T...	Poulet fermier			300g				
5	0000000105	http://world-...	kiliweb	1572117743	2019-10-26T19:22:23Z	1738073570	2025-01-28T...		1740085032	2025-02-20T...	Paleta gran reserva - Sierra nevada-			750ml				
6	00000002	http://world-...	kiliweb	1722606455	2024-08-02T13:47:35Z	1741901006	2025-03-13T...	waistline-app	1741901006	2025-03-13T...	Filets de poulet blanc x2			240-400 g				
7	00000003	http://world-...	prepperapp	1716818343	2024-05-27T13:59:03Z	1741824480	2025-03-13T...	scanada	1741824480	2025-03-13T...	Creatine Gummies			700ml				
8	00000004	http://world-...	elcoco	1560176426	2019-06-10T14:20:26Z	1741442321	2025-03-08T...		1741442321	2025-03-08T...	leite egua			1000ml	Glas	en:glass	Glass	
9	000000475	http://world-...	kiliweb	1714206330	2024-04-27T08:25:30Z	1714207074	2024-04-27T...	roboto-app	1740362538	2025-02-24T...	Confiture extra citron de Menton			230 g				
10	00000005	http://world-...	touchette	1605337720	2020-11-14T07:08:40Z	1740322832	2025-02-23T...	foodless	1740322832	2025-02-23T...				1000ml	Verre, barqu...	en:glass,en:t...	Glass,Tray	
11	00000006	http://world-...	maldan	1732037972	2024-11-19T17:39:32Z	1740141279	2025-02-21T...		1740360225	2025-02-24T...				1000ml				
12	00000006666	http://world-...	prepperapp	1709219541	2024-02-29T15:12:21Z	1738680456	2025-02-04T...	scanbot	1740134585	2025-02-21T...	Gemüse - Gurke			1pcs				
13	00000007	http://world-...	smoothie-app	1678803019	2023-03-14T14:10:19Z	1740403543	2025-02-24T...	bodysupport	1740403543	2025-02-24T...				25 gram				
14	00000008	http://world-...	halal-app-ch...	1609862762	2021-01-05T16:06:02Z	1742039579	2025-03-15T...	insectproductadd	1742039579	2025-03-15T...				700ml				
15	00000009	http://world-...	openfoodfac...	1527242583	2018-05-25T10:03:03Z	1741855645	2025-03-13T...	charlesnepote	1741855645	2025-03-13T...	Mission Carb Balance - Flour Tortilla Burrito			700ml				
16	00000009	http://world-...	prepperapp	1677761447	2023-03-02T12:50:47Z	1727982393	2024-10-03T...	maldan	1727982393	2024-10-03T...	Protein Plant Powered Wrap			1pcs				
17	00000010	http://world-...	jeanbono	1476947941	2016-10-20T07:19:01Z	1736848087	2025-01-14T...	prepperapp	1740357244	2025-02-24T...	Zaatar			700ml	Plastic,Cardb...	en:plastic,en...	Plastic,Cardb...	
18	00000011	http://world-...	prepperapp	1718649410	2024-06-17T18:36:50Z	1736511452	2025-01-10T...	prepperapp	1738687083	2025-02-04T...	Pyrat XO Reserve Rum			1pcs				
19	00000012	http://world-...	elcoco	1578683912	2020-01-10T19:18:32Z	1742154313	2025-03-16T...	prepperapp	1742154313	2025-03-16T...	HIGH PROTEIN ICED COFFEE Herbalife		HIGH PROTEI...					
20	00000013	http://world-...	jeff-krab	1553970319	2019-03-30T18:25:19Z	1740935364	2025-03-02T...	roboto-app	1740935364	2025-03-02T...	Collagen Peptide Powder			300 g	pot plastique	pot-plastique	Pot-plastique	
21	00000014	http://world-...	insectproduc...	1710345659	2024-03-13T16:00:59Z	1735998791	2025-01-04T...	teolemon	1740134903	2025-02-21T...	Süßlupinen Mehl			20pcs				
22	00000015	http://world-...	openfoodfac...	1523810594	2018-04-15T16:43:14Z	1741871782	2025-03-13T...	andreofood	1741871782	2025-03-13T...	Madeleines ChocoLait			1000 g	Plastique, Ca...	en:plastic,en...	Plastic,Cardb...	
23	00000016	http://world-...	maldan	1703127173	2023-12-21T02:52:53Z	1740316739	2025-02-23T...	foodvisor	1740316739	2025-02-23T...	Desiccated liver							
24	00000017	http://world-...	foodvisor	1728736811	2024-10-12T12:40:11Z	1741557289	2025-03-09T...	fmar	1741557289	2025-03-09T...	Whey Protein - Salty Caramel			1.0 kg				
25	00000018	http://world-...	foodvisor	1729882037	2024-10-25T18:47:17Z	1734701994	2024-12-20T...	prepperapp	1738849973	2025-02-06T...	Chocolate peanut butter protein							
26	00000019	http://world-...	prepperapp	1718717714	2024-06-18T13:35:14Z	1729700894	2024-10-23T...	foodvisor	1738828660	2025-02-06T...	Erdbeeren			250g				
27	00000020	http://world-...	openfoodfac...	1536930846	2018-09-14T13:14:06Z	1742152487	2025-03-16T...	waistline-app	1742152487	2025-03-16T...	Organic Buttermilk Powder			1080 g / 50...	1 boîte en ca...	fr:1-boite-en...	fr:1-boite-en...	

File: C:\Users\sahit\...\en.openfoodfacts.org.products (1).csv | 206×37,49,136 | 206×37,49,136 | 18

Figure II: Unrefined Dataset from OpenFoodFacts

All this data was extracted from the Open Food Facts website from the India section [?]. The data was taken from the India section of the website which was sorted by the country sorting option given on the website.

The data on the website had many discrepancy, like some of the packaged food products didn't have the ingredients mentioned in them some didn't have nutrition label, so these entries were injured and out of the 39,00,000 food products in the file around 7000 to 8000 food products were taken into accord and converted into an excel database, containing Packaged product name, Nutri score of each product, barcode of each Food product and the ingredients of all the packaged food products, all this information available on the website itself. Then the data was processed and refined, mainly deletion of some food ingredients that are not so important and only considering the food ingredients that make up most of the packaged food product.

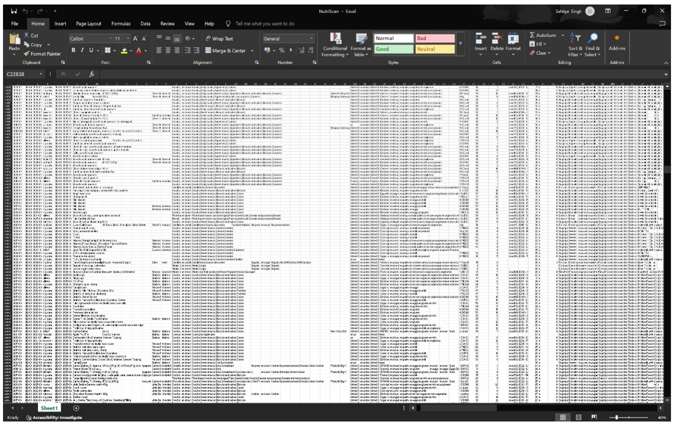


Figure III: Refined dataset of more than 22,000 products

Then this database was imported into the python program in the form of an .csv file(Comma-separated values (CSV) is a text file format that uses commas to separate values, and newlines to separate records. Label encoder was used to transform data entries into numeric form, which was imported from the sklearn library.

A. Algorithms

This Database is then utilized to apply different Machine Learning based models, Random Forest Classification, SVM(Support Vector Machine), Decision Tree Classifier and Logistic Regression.Random forest Classifier is developed by utilizing an estimator value of 100, i.e., there are 100 decision trees from which the model will execute and make decisions based on majority. Class weight was kept as balanced and random state as 1.

Then the Logistic Regression model was made from the same database, the iterations of the model was set to be 1000, iterations are the number of times the model is to be run to find out the best possible result, the class weight was set to be balanced and the random state was set to be 1(to avoid repetition and used to shuffle).

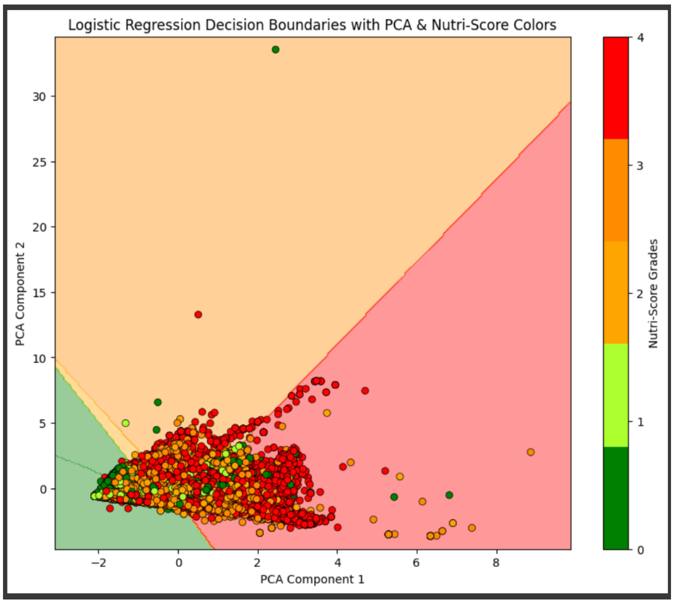


Figure IV: Logistic Regression

SVM with Linear Kernel: An SVM model was created with the kernel type set to 'linear'. The class weight parameter was set to 'balanced', similar to the Logistic Regression model. This parameter scales the weights inversely proportional to class frequencies in the input data.

SVM with Polynomial Kernel: Another SVM model was built by setting the kernel type as 'poly'. Besides changing class weight to 'balanced', degree for the polynomial kernel was set as 3. The degree argument specifies the order of the polynomial kernel function.

SVM with RBF Kernel: A third Support Vector Machine (SVM) model was constructed, using the kernel type known as 'rbf' (Radial Basis Function). The class weight parameter was again set to 'balanced' for this specific model. The RBF kernel is commonly known as the most used kernel and can handle non-linear data.

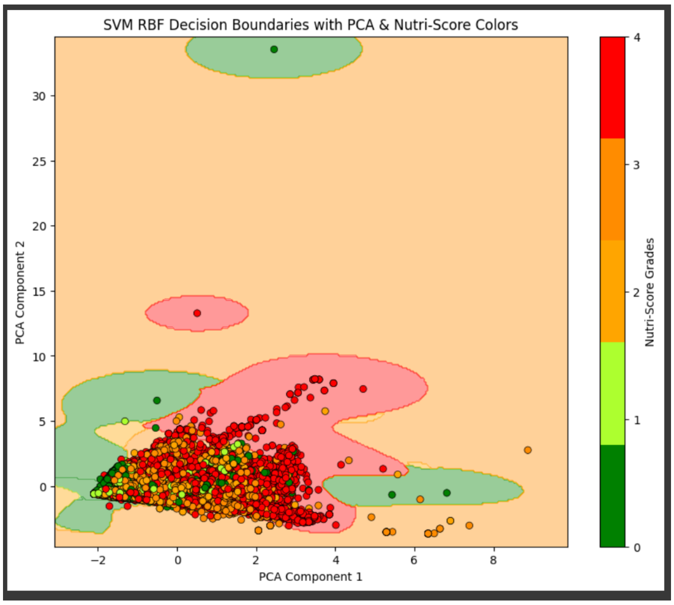


Figure V: SVM RBF Kernel Graph

Decision Tree model was also built from the same database. The random state parameter was passed as 42 for reproducibility, and the class weight parameter was passed as 'balanced' to balance the potential class imbalance in the target variable

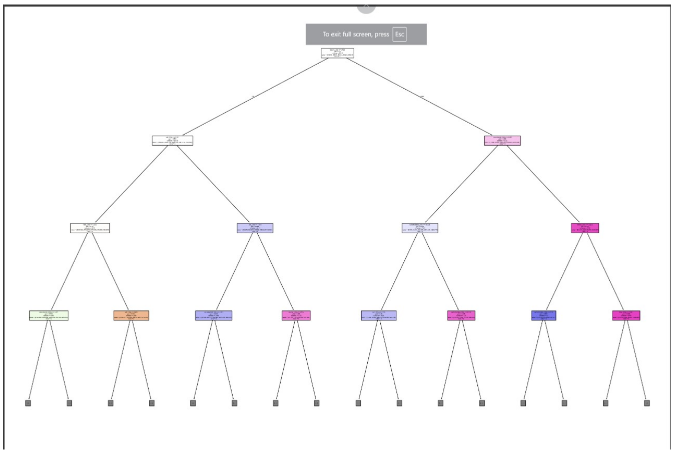


Figure VI: Decision Tree Graph

V. Result

A. Logistic Regression

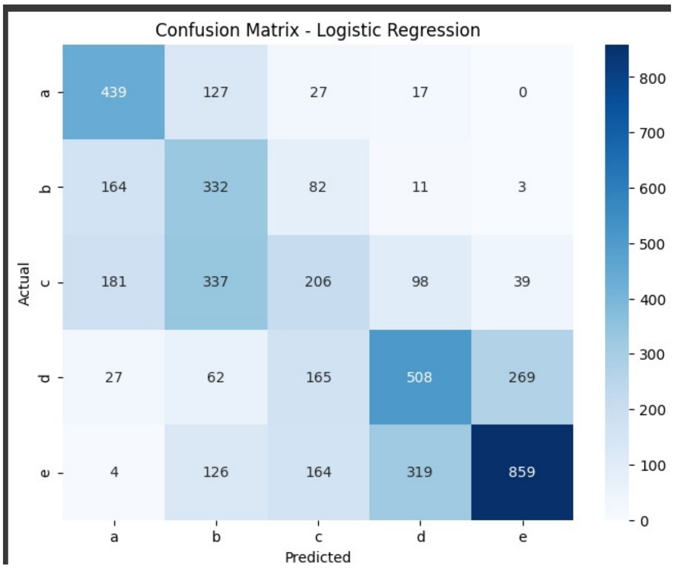


Figure VII: Confusion Matrix Logistic Regression

To gain deeper insights into the classification behavior of the Logistic Regression model, a confusion matrix was generated (refer figure VII). This matrix provides a detailed breakdown of the model's predictions, illustrating the distribution of correct and incorrect classifications across the five designated classes ('a' through 'e').

The diagonal elements of the matrix represent the count of instances correctly classified for each respective class. The model demonstrates the highest number of true positives for class 'e' (859), followed by class 'd' (508). Class 'a' shows the next highest count of true positives (439), followed by class 'b' (332). Class 'c' has by far the lowest number of correctly classified instances (206). These figures indicate the model performs best on class 'e' but struggles significantly with identifying class 'c'.

Examination of the off-diagonal elements reveals specific patterns of misclassification. The most severe confusion occurs where instances belonging to class 'c' are very frequently misclassified as class 'b' (337 instances). Another major error is the misclassification of true class 'e' instances as class 'd' (319 instances). Significant confusion also exists where true class 'd' instances are predicted as class 'e' (269 instances).

Further substantial misclassification trends include:

True class 'c' instances being predicted as class 'a' (181 instances). True class 'b' instances being predicted as class 'a' (164 instances). True class 'd' instances being predicted as class 'c' (165 instances). True class 'e' instances being predicted as class 'c' (164 instances). True class 'a' instances being predicted as class 'b' (127 instances). True class 'e' instances being predicted as class 'b' (126 instances). These specific error types drastically impact the precision and recall metrics for individual classes. For example:

The extremely high misclassification of class 'c' (particularly as 'b' and 'a') results in a very poor recall for class 'c' and significantly diminishes the precision for classes 'b' and 'a'. The strong bidirectional confusion between classes 'e' and 'd' negatively affects the recall and precision for both of these classes. The notable bidirectional confusion between 'a' and 'b' impacts the recall and precision metrics for these classes. The misclassification of 'd' and 'e' instances as 'c' lowers the recall for 'd' and 'e' while also reducing the already low precision of class 'c'. The tendency for 'e' to be misclassified as 'b' further harms the recall of 'e' and the precision of 'b'.

Table I: Classification Report for Logistic Regression Model

Class	Precision	Recall	F1-Score	Support
0	0.54	0.72	0.62	610
1	0.34	0.56	0.42	592
2	0.32	0.24	0.27	861
3	0.53	0.49	0.51	1031
4	0.73	0.58	0.65	1472
Macro Average	0.49	0.52	0.49	4566
Weighted Average	0.53	0.51	0.51	4566

B. SVM Linear Kernel

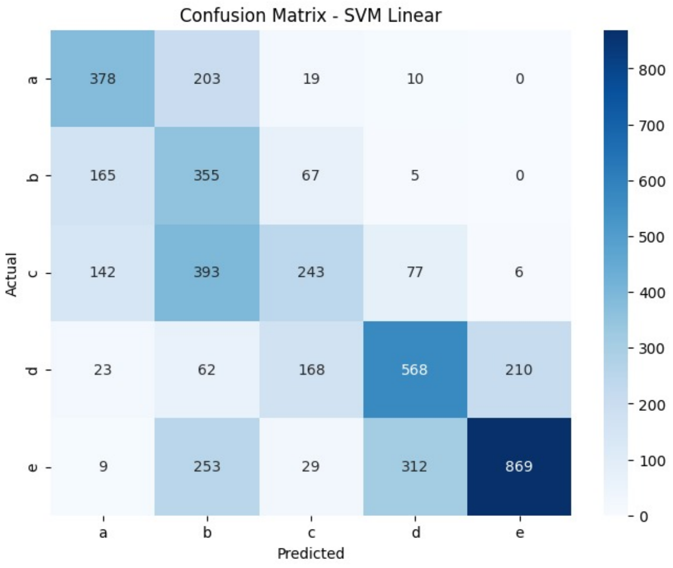


Figure VIII: Confusion Matrix SVM Linear

To gain deeper insights into the classification behavior of the Support Vector Machine model utilizing a Linear kernel (SVM Linear), a confusion matrix was generated (refer figure VIII). This matrix provides a detailed breakdown of the model's predictions, illustrating the distribution of correct and incorrect classifications across the five designated classes ('a' through 'e').

The diagonal elements of the matrix represent the count of instances correctly classified for each respective class. The model demonstrates the highest number of true positives for class 'e' (869), followed by class 'd' (568). Class 'a' shows the next highest count of true positives (378), followed by class 'b' (355). Class 'c' has the lowest number of correctly classified instances (243). These figures suggest the model is most successful in identifying instances of class 'e' and least successful for class 'c'.

Examination of the off-diagonal elements reveals specific patterns of misclassification. The most significant confusion occurs where instances belonging to class 'c' are overwhelmingly misclassified as class 'b' (393 instances). Another very prominent error is the misclassification of true class 'e' instances as class 'd' (312 instances). Furthermore, a large number of true class 'e' instances are also predicted as class 'b' (253 instances).

Other substantial misclassification trends include:

True class 'd' instances being predicted as class 'e' (210 instances). True class 'a' instances being predicted as class 'b' (203 instances). True class 'b' instances being predicted as class 'a' (165 instances). True class 'd' instances being predicted as class 'c' (168 instances). True class 'c' instances being predicted as class 'a' (142 instances). These specific error types directly impact the precision and recall metrics for individual classes. For example:

The extremely high misclassification of class 'c' as 'b' drastically reduces the recall for class 'c' and severely diminishes the precision for class 'b'. The significant confusion between classes 'e' and 'd' (in both directions, but notably 'e' predicted as 'd') impacts the recall and precision for both 'e' and 'd'. The frequent misclassification of 'e' as 'b' further lowers the recall for 'e' and precision for 'b'. The notable bidirectional confusion between 'a' and 'b' reduces both recall and precision for these two classes. The misclassification of 'd' as 'c', and 'c' as 'a', contributes to lower recall for 'd' and 'c' respectively, and lower precision for 'c' and 'a'.

Table II: Classification Report for SVM Linear Model

Class	Precision	Recall	F1-Score	Support
0	0.53	0.62	0.57	610
1	0.28	0.60	0.38	592
2	0.46	0.28	0.35	861
3	0.58	0.55	0.57	1031
4	0.80	0.59	0.68	1472
Macro Average	0.53	0.53	0.51	4566
Weighted Average	0.58	0.53	0.54	4566

C. SVM Polynomial Kernel

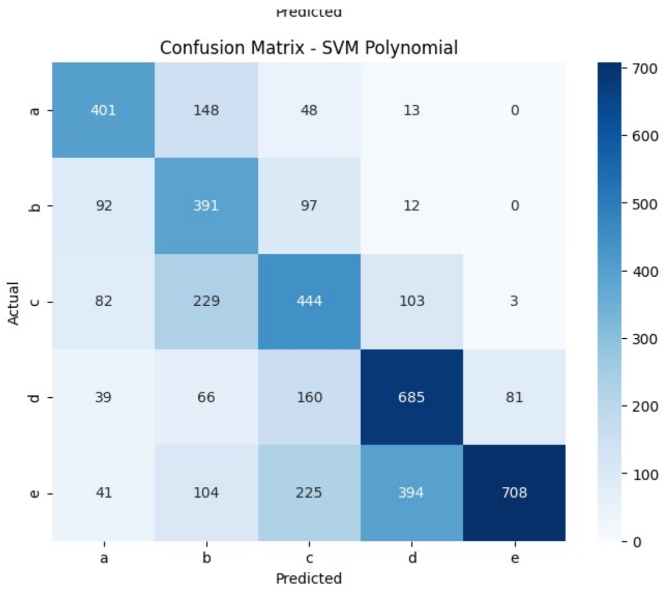


Figure IX: Confusion Matrix SVM Polynomial

To gain deeper insights into the classification behavior of the Support Vector Machine model utilizing a Polynomial kernel (SVM Polynomial), a confusion matrix was generated (refer figure IX). This matrix provides a detailed breakdown of the model's predictions, illustrating the distribution of correct and incorrect classifications across the five designated classes ('a' through 'e').

The diagonal elements of the matrix represent the count of instances correctly classified for each respective class. The model demonstrates the highest number of true positives for class 'e' (708), followed closely by class 'd' (685). Class 'c' also shows a relatively high number of true positives (444), while classes 'a' and 'b' have slightly fewer (401 and 391, respectively). These figures suggest the model is generally most successful in identifying instances of classes 'e' and 'd'.

Examination of the off-diagonal elements reveals specific patterns of misclassification. The most significant confusion is observed where instances belonging to class 'e' are frequently misclassified as class 'd' (394 instances). Another very prominent error occurs when true class 'c' instances are predicted as class 'b' (229 instances). Additionally, a substantial number of true class 'e' instances are predicted as class 'c' (225 instances).

Further notable misclassifications include:

True class 'd' instances being predicted as class 'c' (160 instances). True class 'a' instances being predicted as class 'b' (148 instances). True class 'e' instances being predicted as 'b' (104 instances). True class 'c' instances being predicted as 'd' (103 instances). Confusion also exists between classes 'b' and 'c' (97 true 'b' instances predicted as 'c') and between 'b' and 'a' (92 true 'b' instances predicted as 'a'). These specific error types directly impact the precision and recall metrics for individual classes. For example:

The substantial misclassification of class 'e' as 'd' significantly reduces the recall for class 'e' and the precision for class 'd'. The frequent misclassification of class 'c' as 'b' contributes to a lower recall for class 'c' and diminished precision for class 'b'. The confusion where 'e' is predicted as 'c' lowers recall for 'e' and precision for 'c'. Similarly, the misclassification of 'd' as 'c' affects the recall of 'd' and the precision of 'c'. The confusion between 'a' and 'b' impacts the respective precision and recall values for these classes.

Table III: Classification Report for SVM Polynomial Model

Class	Precision	Recall	F1-Score	Support
0	0.61	0.66	0.63	610
1	0.42	0.66	0.51	592
2	0.46	0.48	0.48	861
3	0.57	0.66	0.61	1031
4	0.89	0.48	0.63	1472
Macro Average	0.59	0.60	0.57	4566
Weighted Average	0.64	0.58	0.58	4566

D. SVM RBF Kernel

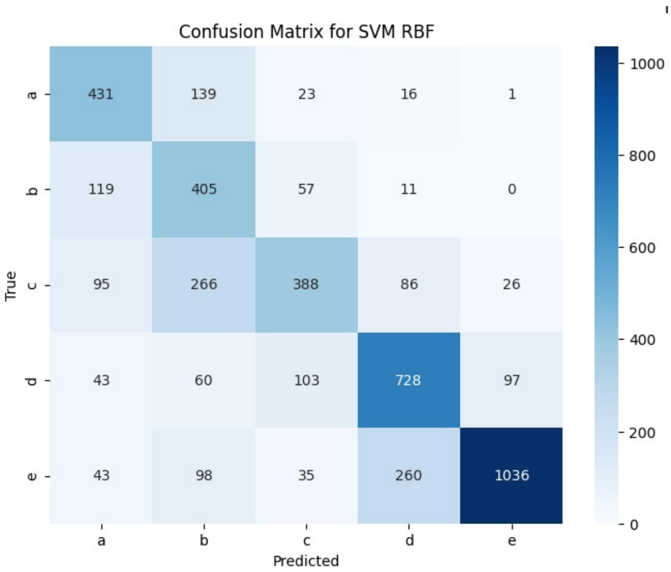


Figure X: Confusion Matrix SVM RBF

To gain deeper insights into the classification behavior of the Support Vector Machine model utilizing a Radial Basis Function (SVM RBF) kernel, a confusion matrix was generated (refer figure X). This matrix provides a detailed breakdown of the model's predictions, illustrating the distribution of correct and incorrect classifications across the five designated classes ('a' through 'e').

The diagonal elements of the matrix represent the count of instances correctly classified for each respective class. The model demonstrates the highest number of true positives for class 'e' (1036), followed by class 'd'

(728). Classes ’a’, ’b’, and ’c’ show progressively fewer true positives (431, 405, and 388, respectively). These figures generally align with the F1-scores presented in Table IV, confirming the relative success in identifying instances of classes ’e’ and ’d’.

Examination of the off-diagonal elements reveals specific patterns of misclassification. A significant confusion is observed where instances belonging to class ’c’ are frequently misclassified as class ’b’ (266 instances). Another prominent error occurs when true class ’e’ instances are predicted as class ’d’ (260 instances). Furthermore, notable confusion exists between classes ’a’ and ’b’, with 139 true ’a’ instances predicted as ’b’, and 119 true ’b’ instances predicted as ’a’.

Additional misclassification trends include instances of class ’c’ being predicted as ’a’ (95) and ’d’ (86), instances of class ’d’ being predicted as ’c’ (103), and instances of class ’e’ being predicted as ’b’ (98). These specific error types directly impact the precision and recall metrics observed for individual classes in the classification report (Table IV). For example, the substantial misclassification of class ’c’ as ’b’ contributes to the reduced recall for class ’c’ and the diminished precision for class ’b’. Similarly, the confusion between ’d’ and ’e’ affects the respective precision and recall values for these classes.

Table IV: Classification Report for SVM RBF Model

Class	Precision	Recall	F1-Score	Support
0	0.59	0.71	0.64	610
1	0.42	0.68	0.52	592
2	0.64	0.45	0.53	861
3	0.66	0.71	0.68	1031
4	0.89	0.70	0.79	1472

Macro Average 0.64 0.65 0.63 4566
Weighted Average 0.69 0.65 0.66 4566

E. Random Forest Classifier

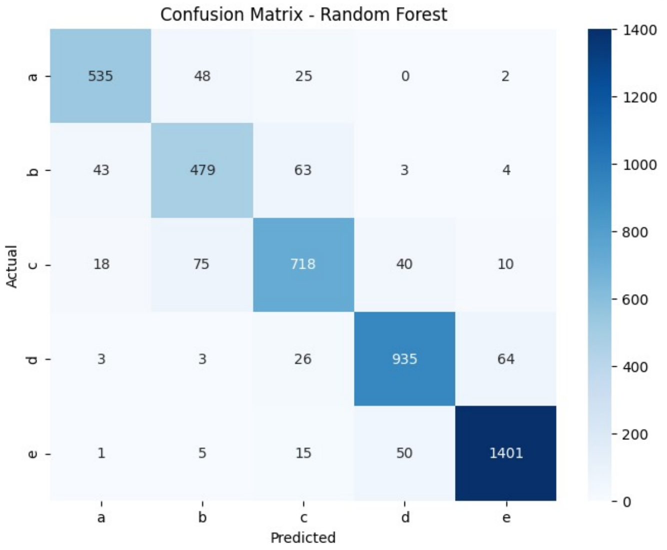


Figure XI: Confusion Matrix Random Forest Classifier

This section details the performance evaluation of a Random Forest classification algorithm applied to the five-class dataset (’a’ through ’e’). The efficacy of the model is quantitatively assessed using the confusion matrix presented (refer figure XI), which tabulates the prediction outcomes against the actual class labels.

The analysis of the confusion matrix indicates a high degree of predictive accuracy achieved by the Random Forest classifier. The prominent values along the main diagonal (True Positives: 535 for ’a’, 479 for ’b’, 718 for ’c’, 935 for ’d’, 1401 for ’e’) significantly outweigh the off-diagonal misclassification counts. This demonstrates the ensemble model’s strong capability in discerning the underlying patterns within the dataset, resulting in a superior overall accuracy of approximately 89.1% (4068 correct predictions out of 4566 total instances).

Examining class-specific performance reveals the model’s particular strengths. The Random Forest classifier exhibits exceptional performance for class ’e’, achieving a recall of approximately 95.2% (1401 out of 1472 instances correctly identified) and high precision (94.6%). Class ’d’ is also classified with high accuracy (Recall 90.7%, Precision 91.0%), closely followed by class ’a’ (Recall 87.7%, Precision 89.2%). These results underscore the robustness of the Random Forest approach in accurately modeling the feature space for these classes.

Despite the high overall performance, certain classification challenges persist. Similar to observations with simpler tree models, the Random Forest encounters difficulty in definitively separating classes ’b’ and ’c’. Notable mutual confusion remains, with 75 instances of actual class ’c’ being misclassified as ’b’, and 63 instances of actual class ’b’ being misclassified as ’c’. While the recall for both classes has improved compared to a single Decision

Tree (Class 'c' Recall 83.4%, Class 'b' Recall 80.9%), class 'b' continues to exhibit the lowest precision among all classes (78.5%). This suggests that even with the ensemble method, the feature distinctions between 'b' and 'c' remain inherently challenging. Minor confusion also persists between 'd' and 'e' (64 'd' as 'e', 50 'e' as 'd') and 'a' and 'b' (48 'a' as 'b', 43 'b' as 'a').

The dataset's class imbalance (support ranging from 592 for 'b' to 1472 for 'e') appears well-handled by the Random Forest model. It demonstrates robustness by maintaining strong performance across all classes, including those with lower support, while excelling on the most frequent class 'e'.

Table V: Classification Report for Random Forest Model

Class	Precision	Recall	F1-Score	Support
0	0.89	0.88	0.88	610
1	0.79	0.81	0.80	592
2	0.85	0.83	0.84	861
3	0.91	0.91	0.91	1031
4	0.95	0.95	0.95	1472

Macro Average 0.88 0.88 0.88 4566
Weighted Average 0.89 0.89 0.89 4566

F. *Decision Tree Classifier*

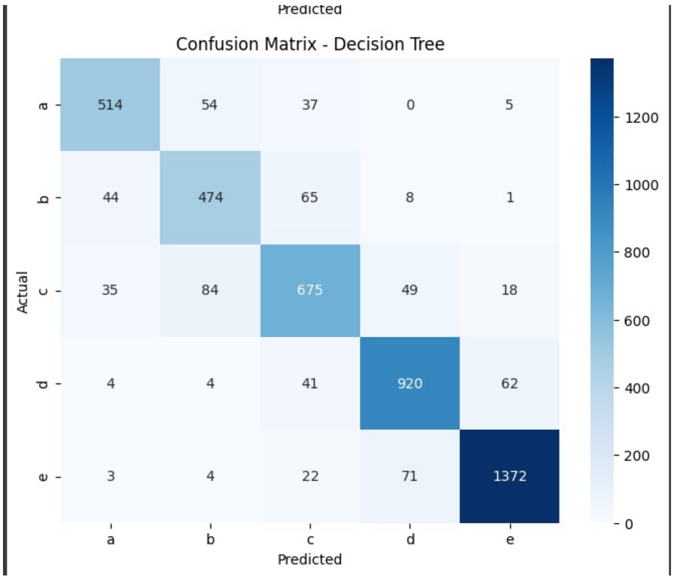


Figure XII: Confusion Matrix Decision Tree Classifier

An evaluation of a Decision Tree classification algorithm was conducted to determine its efficacy on the given dataset involving five distinct classes (labeled 'a' through 'e'). The performance is detailed in the confusion matrix presented (refer figure XII). This matrix visualizes the counts of true positive, false positive, false negative, and true negative predictions for each class.

The analysis reveals that the Decision Tree classifier demonstrates generally strong predictive performance. The diagonal elements of the confusion matrix, representing correct classifications (True Positives: 514 for 'a', 474 for 'b', 675 for 'c', 920 for 'd', 1372 for 'e'), are substantially larger than the off-diagonal elements (misclassifications) across all classes. This indicates that the Decision Tree structure is reasonably well-suited for capturing the underlying patterns differentiating most classes within this dataset, achieving an overall accuracy of approximately 86.6% (3955 correct predictions out of 4566 total instances).

Class-specific analysis highlights variations in the model's effectiveness. The classifier exhibits particularly high efficacy for class 'e', correctly identifying 1372 out of 1472 instances (Recall 93.2%) with minimal confusion from other classes being predicted as 'e'. Class 'd' also shows robust classification (Recall 89.2%), followed by class 'a' (Recall 84.3%). These results suggest the features associated with classes 'e', 'd', and 'a' are more distinct and effectively learned by the Decision Tree.

Conversely, the model encounters significant challenges in discriminating between classes 'b' and 'c'. These two classes exhibit the most substantial mutual confusion: 84 instances of actual class 'c' are misclassified as 'b', while 65 instances of actual class 'b' are misclassified as 'c'. Consequently, class 'c' registers the lowest recall (78.4%), and class 'b' shows relatively lower precision compared to other classes. This pattern suggests potential feature overlap or complexity in the decision boundaries between these specific classes that the current tree structure struggles to resolve optimally. Minor confusion is also observed between classes 'd' and 'e' (62 'd' as 'e', 71 'e' as 'd') and classes 'a' and 'b' (54 'a' as 'b', 44 'b' as 'a').

The dataset appears to exhibit class imbalance, with support ranging from 592 instances (class 'b') to 1472 instances (class 'e'). The Decision Tree demonstrates reasonable robustness, achieving good recall even for the least frequent class ('b' 80.1%), although its strongest performance aligns with the most frequent class ('e' 93.2%).

Table VI: Classification Report for Decision Tree Model

Class	Precision	Recall	F1-Score	Support
0	0.86	0.84	0.85	610
1	0.76	0.80	0.78	592
2	0.80	0.78	0.79	861
3	0.88	0.89	0.89	1031
4	0.94	0.93	0.94	1472

Macro Average 0.85 0.85 0.85 4566
Weighted Average 0.87 0.87 0.87 4566

After constructing and evaluating all the models, the Random Forest Classifier was 89.48751642575559% accurate, whereas the logistic regression was 50.45992115637319% accurate. The accuracies of various Support Vector Machine (SVM) models were the following: Linear Kernel was 53.5041611914148% accurate, Polynomial Kernel was 53.46035917652212% accurate, and RBF Kernel was 61.65133596145422% accurate. Additionally, the Decision Tree Classifier was 86.55278142794569% accurate.

Table VII: Accuracy Table

Model	Accuracy
Logistic Regression	51.3359614542269
SVM Linear	52.84713096802452
SVM Polynomial	57.57774857643452
SVM RBF	65.44021024967148
Random Forest Classifier	89.09329829172142
Decision Tree Classifier	0.8661848445028472

VI. Discussion

The performance of six different classification algorithms was compared to assess their performance on the given dataset. The performance metrics like precision, recall, F1-score, and support for each class as well as macro and weighted averages are presented in Tables I to VI.

A comparative analysis reports significant difference in predictive performance between the models under consideration. Of special interest, tree-structure-based algorithms showed significantly improved performance. The Random Forest (Table V) had the highest performance level with a weighted average F1-score of 0.89 corroborated by a macro average F1-score of 0.88. Likewise, the Decision Tree classifier (Table VI) had impressive performance, with weighted and macro average F1-scores of 0.87 and 0.85, respectively. These results affirm that ensemble methods and decision tree structures are most appropriate to identify the inherent patterns inherent in this dataset.

Conversely, Support Vector Machine (SVM) models performed differently depending on the kernel used. The SVM using a Radial Basis Function (RBF) kernel (Table IV) performed the best of the various SVM configurations, with a weighted average F1-score of 0.66. Then, the Polynomial kernel SVM (Table III) performed at a weighted average F1-score of 0.58. The Linear kernel SVM (Table II) and the Logistic Regression model (Table I) were the poorest level of performance, with weighted average F1-scores of 0.54 and 0.51, respectively. The relatively lower efficiency of linear models and standard SVM kernels indicates that the data contains potential non-linear complexities that these techniques appear not to capture well.

The class imbalance is very severe, with a clear indication from the 'Support' column in all tables ranging from 592 to 1472 instances per class. The top two models, Random Forest and Decision Tree, were not affected by the imbalance to a large degree, with a very high F1-score being obtained consistently in all classes while showing negligible difference between their macro and weighted average scores. For all the remaining models, the performance was sensitive to the differences in class distribution. For Logistic Regression, SVM Linear, and SVM RBF, the weighted averages were marginally higher than the macro averages, reflecting a relatively better performance on the more common classes.

The class-specific results show uniform struggles. Class 2 was the most challenging category to classify for the majority of models, registering the lowest F1-scores, particularly significant for Logistic Regression (0.27) and SVM Linear (0.35). Even leading models had relatively lower (but still high) F1-scores for Class 2 (Random Forest: 0.84, Decision Tree: 0.79) than the other classes. Class 4 was typically the highest-performing class, recording F1-scores of as high as 0.95 with the Random Forest model.

In general, empirical data strongly demonstrates that the tree-structure-based classifiers, namely Random Forest, yield the optimal solution to this classification task with improved accuracy and efficient management of class imbalance compared to the examined linear and SVM approaches.

VII. Conclusion

The primary objective of this research was to develop a packaged food labeling model capable of assigning nutrition scores ranging from A to E based on a product's ingredient composition. To achieve this, machine learning algorithms — specifically, Random Forest Classifier and Linear Regression — were employed and evaluated side by side for their effectiveness. Experimental results indicate that both models perform comparably well on smaller datasets (ranging from 500 to 700 records), demonstrating reliable classification capabilities even with limited data. This reinforces the potential of machine learning as a practical tool for nutritional assessment and consumer awareness. The developed model provides a systematic approach for evaluating the healthiness of packaged food items, enabling consumers to make informed dietary choices. By clearly labeling products based on nutritional quality, the system encourages healthier consumption habits and can serve as a valuable aid in promoting public health. Future work involves expanding the dataset from approximately 700 to over 2000 entries to enhance the model's accuracy and generalizability. Furthermore, deploying the model as a web-based or mobile application can significantly improve its accessibility and usability, allowing users to effortlessly assess products and make data-driven decisions to improve their nutritional well-being.

References

[1] Ross A.C., Caballero B., Cousins R.J., Tucker K.L. Modern Nutrition in Health and Disease. Jones and Bartlett Learning; Burlington, MA, USA: 2020. [Google Scholar]

[2] Whitney E.N., Rolfes S.R., Crowe T., Walsh A. Understanding Nutrition. Cengage; Melbourne, Australia: 2019. [Google Scholar]

[3] Melaku Y.A., Temesgen A.M., Deribew A., Tessema

G.A., Deribe K., Sahle B.W., Abera S.F., Bekele T., Lemma F., Amare A.T., et al. The impact of dietary risk factors on the burden of non-communicable diseases in Ethiopia: Findings from the Global Burden of Disease study 2013. Int. J. Behav. Nutr. Phys. Act. 2016;13:122. doi: 10.1186/s12966-016-0447-x. [DOI] [PMC free article] [PubMed] [Google Scholar]

[4] Popkin B.M., Adair L.S., Ng S.W. Global nutrition transition and the pandemic of obesity in developing countries. Nutr. Rev. 2012;70:3–21

[5] Mozaffarian D. Dietary and policy priorities for cardiovascular disease, diabetes, and obesity: A comprehensive review. Circulation. 2016;133:187–225. doi: 10.1161/CIRCULATIONAHA.115.018585. [DOI] [PMC free article] [PubMed] [Google Scholar]

[6] Awuchi C.G., Igwe V.S., Amagwula I.O. Nutritional diseases and nutrient toxicities: A systematic review of the diets and nutrition for prevention and treatment. Int. J. Adv. Acad. Res. 2020;6:1–46. doi: 10.46654/ij.24889849.e61112. [DOI] [Google Scholar]

[7] Goyal A., Gupta Y., Singla R., Kalra S., Tandon N. American diabetes association "standards of medical care—2020 for gestational diabetes mellitus": A critical Appraisal. Diabetes Ther. 2020;11:1639–1644. doi: 10.1007/s13300-020-00865-3. [DOI] [PMC free article] [PubMed] [Google Scholar]

[8] Pray L., editor. Nutrigenomics and the Future of Nutrition: Proceedings of a Workshop. National Academies Press; Washington, DC, USA: 2018. [PubMed] [Google Scholar]

[9] Jimenez-Carvelo A.M., Cuadros-Rodríguez L. Data mining/machine learning methods in foodomics. Curr. Opin. Food Sci. 2021;37:76–82. doi: 10.1016/j.cofs.2020.09.008. [DOI] [Google Scholar]

[10] Taye M.M. Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. Computers. 2023;12:91. doi: 10.3390/computers12050091. [DOI] [Google Scholar]

[11] https://matplotlib.org/stable/users/index

[12] https://scikit-learn.org/

[13] https://numpy.org/doc/

[14] https://pandas.pydata.org/docs/