

# Foods for the Goals: A Comparative Study of Machine Learning Algorithms for Fitness Goals-Based Food Classification

Quiel Andrew I. Quiwa  
College of Engineering  
University of the Philippines Diliman  
Quezon City, Philippines  
qiqiwa@up.edu.ph

Joel Paolo C. Salvador  
College of Engineering  
University of the Philippines Diliman  
Quezon City, Philippines  
jcsalvador3@up.edu.ph

## ABSTRACT

This study investigates and compares four machine algorithms: Naive Bayes, Multilayer Perceptron, Support Vector Machines, and Adaboost to classify different types of food based on the following fitness goals: (1) muscle gain, (2) endurance, and (3) weight loss, using a dataset containing macro- and micronutrient profiles, with a threshold-based labeling based on nutrition literatures.

The four models were trained and evaluated using 5-fold cross-validation, and was compared using their performance metrics (i.e. accuracy, precision, recall, f1-score), with f1-score selected as the primary metric. The models performed relatively well with scores around 90% but Adaboost performed with the highest f1-score.

Beyond classification, this study also conducted exploratory analyses using a one-way ANOVA ( $p < 0.5$ ) and logistic regression as a tool to interpret results. The significant features were then added to the feature space to determine whether the model will perform better when loaded with additional important features.

## 1. PROBLEM STATEMENT

Food consumption is a critical factor affecting an individual's level of fitness. While people have varying fitness goals depending on their lifestyle or daily activities, the quality and quantity of food consumed can influence how well those goals are achieved. Unfortunately, access to information on which types of food best support specific fitness goals is limited and eating healthy is often associated with higher costs.

This project explores different classifier algorithms to classify food into three fitness labels: (1) Muscle gain which is associated with high protein intake for caloric surplus; (2) Weight loss which is associated with reduced carbohydrate and fat intake to achieve caloric deficit; and (3) Endurance which is focused on high carbohydrate diet to maximize energy reserves for long exercises.

Given these labels: three primary features will be used to classify food: (1) protein, (2) fat, (3) carbohydrate, (4) sodium, (5) sugar, (6) caloric value, and (7) dietary fiber. Adjacently, the project will also explore other macronutrient trends of food within each fitness label to better under-

stand the characteristics that differentiate these categories. Through this classifier, the project aims to improve access to information on food options that match a person's fitness goal. Potential use cases in the future can also be expanded towards addressing macronutrient deficiencies, as well as specific dietary restrictions of individuals.

## 2. RELATED LITERATURE

One of the barriers to eating healthy, as identified by Solan, is the misguided assumptions of people, who usually view eating healthy as being too expensive and too difficult to prepare in order for them to adopt nutritious diets [1]. Nutrition plans, including macronutrient breakdowns, should be tailored to each individual, depending on multiple factors, including fitness goals [2].

For muscle gain, the recommended daily protein intake is at least 2.2g protein/kg of an individual's weight [3]. For weight loss, a daily caloric deficit of 500 kcals is identified to reduce body fat while preserving muscle [4]. For endurance, about 7-12g carbohydrates/kg of an individual's weight is required for heavy training [5].

Similar use cases have recently been explored by machine learning researchers. Al Qohar, et. al classified Indonesian food based on their nutritional profiles using KNN, SVM, random forest, and multi-layer perceptron (MLP) [6]. Senthilmurugan, et. al, evaluated a SVM classifier for fast food nutrition data [7]. Ajami and Teimourpour developed a food recommender system using AdaBoost [8]. Ahuja, et. al developed a food classification algorithm using KNN and Naive Bayes [9].

Overall, this project leverages on these literatures to improve access to information on fitness goals-based food, while also diving on the macronutrient trends among the foods within the fitness classes.

## 3. DATASET

The dataset used was the Food Nutrition Dataset by Ut-sav Dey on Kaggle. It is a comprehensive database of different foods containing information about their macro and micronutrients.

In addition, all other features, including the three mentioned

previously, was used to identify the macronutrient trends of foods within each fitness class.

## 4. METHODOLOGY

### 1. Data Pre-processing

For data pre-processing, feature columns were first identified to ensure that only relevant nutritional attributes were included in the model. The features identified were primarily macronutrients, particularly protein, caloric value, carbohydrates, sugars, sodium, dietary fiber, and fat. These are the features that were used to assign the labels for the entries. Later in the experiment, additional feature columns were also included to test the model with additional features but were not used in labeling.

After the feature columns for consideration were decided, an exploratory data analysis was conducted using the pandas library to gain an initial understanding of the dataset. This included checking for the shape of the dataset, dropping rows with empty values for the selected features, and assigning the dataframes for the features and labels to their respective matrix variables. Some basic EDA statistics were also explored by describing the dataframe to gain a general understanding of dataset:

**Table 1: Descriptive Statistics of Key Nutrients**

	Protein	Fat	Carbs	Caloric Value
count	2395.0000	2395.0000	2395.0000	2395.0000
mean	13.4007	10.1762	18.5890	223.7695
std	32.2942	29.0089	29.4061	384.7282
min	0.0000	0.0000	0.0000	0.0000
25%	0.8000	0.3000	0.5000	44.5000
50%	3.5000	2.1000	6.8000	117.0000
75%	13.3000	9.4000	25.0500	258.0000
max	560.3000	550.7000	390.2000	6077.0000

The values above assured the project team that the dataset is generally within the acceptable and expected ranges before proceeding with model development.

### 2. Threshold-based labeling

Since the objective of the project is to compare classifiers using supervised learning methods, and the dataset used did not initially have labels, label assignment rules were applied on each row based on studies related to macronutrient features gathered from nutrition-based journal articles. Although the project recognizes that the best results can be achieved through proper consultation with a certified or professional nutritionist, the labels were decided with the following logic:

1. The rows were labeled as "Muscle Gain" if the food macronutrients contained Protein  $\geq 20$ g, or if Protein  $\geq 15$  g and Calories  $\geq 250$  kcal. Nutrition research articles suggest that Muscle Protein Synthesis rates or the process

that turns muscle into protein were maximized by ingesting a 20g dose of whey protein after a protein-rich breakfast [10]. Additionally, there is research showing that MPS can be maximized in young adults with an intake of 20-25g of high quality protein [11]. Caloric value was also included in this label since several journals have referenced that energy surplus is required to facilitate muscle hypertrophy [12]. Although no exact value has been determined, the project selected anything greater than 250kcal despite having only about 15g protein could still be labeled as food for muscle gain.

2. The rows were labeled as "Endurance" if the food macronutrient values were Carbohydrates  $\geq 30$  g, or Sugars  $\geq 12$  g, or (Carbohydrates  $\geq 20$ g and Sodium  $\geq 140$  mg). This was based on articles citing endurance requiring 30-60 g of carbohydrate intake for endurance events and regular ingestion of sodium beverages for proper hydration [13]. Sugars such as glucose, sucrose, or carbohydrates that are simple sugars were also identified to contribute to endurance so this was also selected in the criteria [14]. This is why food which were not above 30g of carbohydrates, but had significant amounts of sugars and sodium were also included in the labeling rules.

3. Lastly, the rows were labeled as "weight loss" if the food macronutrient values were Calories  $< 150$  kcal, Fat  $< 5$  g, and (Carbohydrates  $< 20$  g or Dietary Fiber  $\geq 3$  g or Protein  $\geq 10$ g). Lower calorie intake is a requirement to achieve caloric deficit for weight loss, and 150kcal is a typical value for smaller portions of snacks and general food items. However, other macronutrient thresholds were also identified as requirements for this deficit to be considered healthy for weight loss. A study found that combining increased fruit and vegetable intakes with decreased fat intake, is an effective strategy for managing body weight while controlling hunger [15]. Additionally, there are also studies showing that fiber intake is associated with lower body weight [16]. Hence, an added requirement for this label is to have lower fat content and either one of having low carbohydrates, high dietary fiber, and relatively higher protein. Across all of these additional criteria, the low calorie measure remains a requirement.

The python-based code logic for setting the threshold explained is shown below:

```
def assign_label(row):
    # Muscle Gain
    if row["Protein"] >= 20 or (row["Protein"]
                               >= 15 and row["Caloric_Value"] >= 250):
        return "Muscle_Gain"

    # Endurance
    if row["Carbohydrates"] >= 30 or
       row["Sugars"] >= 12 or \
       (row["Carbohydrates"] >= 20 and
        row["Sodium"] >= 140):
        return "Endurance"

    # Weight Loss
    if (row["Caloric_Value"] < 150) and
```

```

        (row["Fat"] < 5) and \
        (row["Carbohydrates"] < 20 or
         row["Dietary_Fiber"] >= 3 or
         row["Protein"] >= 10):
            return "Weight_Loss"

# Default
return "No_Category"

```

Value counts were implemented to ensure that the threshold based labeling for the dataset included enough values for each label:

**Table 2: Label Distribution Counts**

Label	Count
Weight Loss	1040
Muscle Gain	500
Endurance	473
No Category	382

### 3. Standardization for Model Comparison

To ensure that the models are compared properly and accordingly, a standardized model development procedure was applied across the different algorithms. **GridSearchCV** was used, except for Naive Bayes, to identify the best combination of the parameters before each model was evaluated.

All models were trained and assessed using a **5-fold cross-validation**, which splits the dataset into 5 equal parts, and produces 5 separate performance scores. This 5-fold cross-validation ensures that every data point in the dataset is used both for training and testing, which reduces bias. The 5 performance scores generated were then averaged out to get its mean performance.

The metrics computed for each model were as follows: (1) accuracy, (2) precision, (3) recall, and (4) f1-score. These scores provide a complete picture of how well each model performed across the three different categories.

Lastly, **confusion matrices** were also generated for each classifier to help visualize the misclassifications that occurred in each model.

### 4. Libraries Used

**a. SciKit Learn** - The following modules and classes from SciKit Learn were used in this project:

**Table 3: Libraries, Modules, and Classes Used in this Project (Part 1)**

Module	Class	Description
sklearn.ensemble	Adaboost Classifier	Scikit Learn provides libraries for supervised machine learning models that were used in the project.
sklearn.naive_bayes	GaussianNB	
sklearn.svm	SVC	
sklearn.neural_network	MLPClassifier	
sklearn.preprocessing	Standard Scaler; Label Encoder	Standard Scaler standardizes the features by computing the mean and standard deviation of each feature to scale each row. Label encoder converts string labels to integers.
sklearn.model_selection	GridSearchCV; train_test_split; cross_val_score	GridSearchCV iterates across all parameters defined to identify the best combination. Train_test_split establishes stratification to maintain class distribution in train and test sets to keep ratio of classes. cross_val_score computes the F1 macro score of each combination of parameters.
sklearn.metrics	accuracy_score; f1_score; classification_report; confusion_matrix; ConfusionMatrixDisplay	These were the output metrics used to compare the model performances in the project.
sklearn.pipeline	Pipeline	This was used to ensure the correct order of the workflow.

**b. Pandas** – This library was primarily used for exploratory data analysis, including reading the dataset, generating initial insights out of the datasets, and providing tabular visualizations on the data.

**c. Matplotlib Library** - This was used to generate output data visualizations, mainly the confusion matrix plot.

**d. Seaborn** - This library was also used to generate the necessary data visualizations on top of the matplotlib library.

**e. OS and Glob** - This library was used to load the dataset.

## 5. Implementation of Four Classifier Models

### a. Naive Bayes Classifier

For the Naive Bayes Classifier, Gaussian Naive Bayes was used as the algorithm in this project since nutritional components food in the dataset are continuous numerical values. Naive Bayes, unlike other models, does not require tuning so no parameter optimization using GridSearchCV was performed for this model.

In this project, the Naive Bayes mainly served as a baseline for comparison of the different models. Because the algorithm relies on strong independence assumptions between features, it generally performs well on simple, cleanly separated data. However, Naive Bayes is less flexible than the other three algorithms. As such, this served as the baseline for improvement comparison of the other advanced models (Support Vector Machines, Multilayer Perceptron, and AdaBoost) over a straightforward Naive Bayes classifier.

```
# Setting of Gaussian Naive Bayes
# as the NB Algorithm to be used

nb_model = GaussianNB()
```

### b. Support Vector Machines

In this project, a Support Vector Machine classifier was also explored. Given that nutritional data do not form clean linear boundaries, the SVM model used Radial Basis Function (RBF) as the kernel. RBF is useful since it allows non-linearly separable variables to be mapped into a higher-dimensional space where classification becomes easier.

Two parameters were tuned for SVM, namely, C and gamma. C controls how strict or flexible the margin is, and gamma determines how far the influence of each training point reaches. These parameters shape the decision boundary of the RBF kernel.

To tune the parameters, GridSearchCV was used to test different combinations of the values of  $c = 0.1, 1, 10$ , and of the values of  $\gamma = 0.01, 0.1, 1$ , with the goal of finding the combination of the parameters that yields the highest macro-averaged F1 score.

The best combination of  $c$  and  $\gamma$  from the grid search was then used to evaluate the final SVM model.

```
#SVM Pipeline -- scaling and classifier
svm_model = Pipeline([
    ("scaler", StandardScaler()),
    ("svc", SVC(kernel="rbf", random_state=42))
])

#Grid Search
param_grid = {
    "svc_C": [0.1, 1, 10],
    "svc_gamma": [0.01, 0.1, 1]
}

grid_search = GridSearchCV(
    estimator=svm_model,
```

```
    param_grid=param_grid,
    scoring="f1_macro", #This is the one being
                        optimized
    cv=5 #5-fold cross-validation
)
```

### c. Adaboost Classifier

In addition to Naive Bayes and SVM, an Adaboost Classifier was also used to explore whether using multiple weak learners at different learning rates could improve the classification performance for this nutrition dataset.

Initially, an arbitrary base value of 100 weak learners with the learning rate set to 1 was tested. This was discovered to have a decent performance at Mean F1 = 0.968522. Hence, tuning these parameters involved testing values at the  $\pm 100$  range from the base value of the number of weak learners and  $\pm 1/0.9$  range from the base value of the learning rates. The GridSearchCV class was also used to calculate all possible values of the parameters by iterating across all combinations of the range provided. The best combination of weak learners and learning rates calculated using GridSearchCV was used for the final model

Additionally, this Adaboost Classifier utilized the SAMME algorithm since the SciKit Learn library's default Adaboost algorithm (SAMME.R) has been deprecated. SAMME or "Stagewise Additive Modeling using Multi-class Exponential Loss" is an extension of AdaBoost used to handle problems with more than two classes using the following adjusted computation:

Adaboost weight computation:

```
alpha = 0.5 * log((1 - error) / error)
```

SAMME adjusted weight computation:

```
alpha = log((1 - error) / error) + log(K - 1)
```

Where  $K$  is the number of classes, so that the extra term strengthens the confidence of accurate weak classifiers in multiclass settings. More classes mean each weak learner is less likely to guess correctly by chance. The term  $\log(K-1)$  adjusts for that, rewarding strong learners more.

### d. Multi-Layer Perceptron Classifier

Finally, the project tested the dataset using a Multi-Layer Perceptron Classifier. In this case, up to three hidden layers of backpropagation were initially tested using SciKit Learn's MLPClassifier class. The methodology did not explore more than three hidden layers since the result of the model at two hidden layers was already found to be decent, and increasing the hidden layers to three no longer resulted in significant improvement on the model.

The neuron sizes for each hidden layer here were tested following a range of 50 to 200 neurons. The alpha values were tested on different ranges from 0.0001 to 0.01. The learning rates were tuned from 0.0005 to 0.0025. Once again, the GridSearchCV class was used to calculate the possible values of the parameters by iterating across all possible combina-

tions of the hidden layer sizes, alpha values, and learning rates within the range and values provided. The best combination was used for the final model.

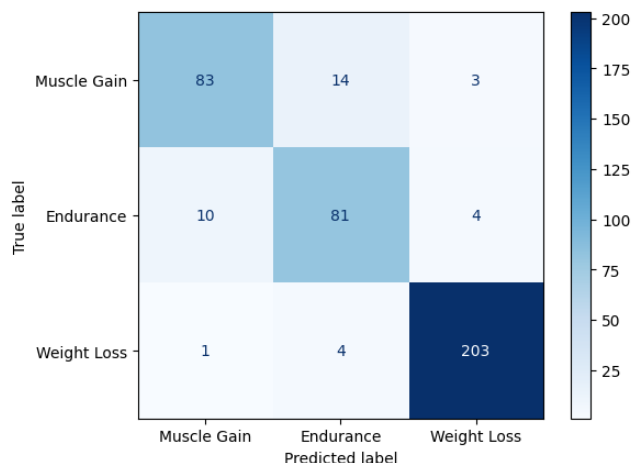
## 6. Results and Discussion

**Table 4: Performance Scores of the Models**

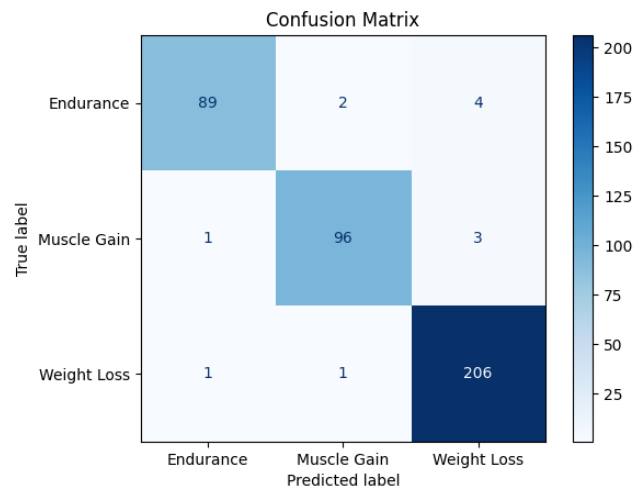
Model	Accuracy	Recall	Precision	F1 Score
Naive Bayes	0.9100	0.8956	0.8944	0.8920
SVM	0.9498	0.9368	0.9452	0.9369
MLP	0.9682	0.9620	0.9624	0.9615
AdaBoost	<b>0.9846</b>	<b>0.9800</b>	<b>0.9873</b>	<b>0.9833</b>

Based on the results, the model that achieved the highest accuracy score of 98.46% was the Adaboost model. It also achieved the highest recall score of 98.00%, precision score of 98.73%, and F1-score of 98.33%

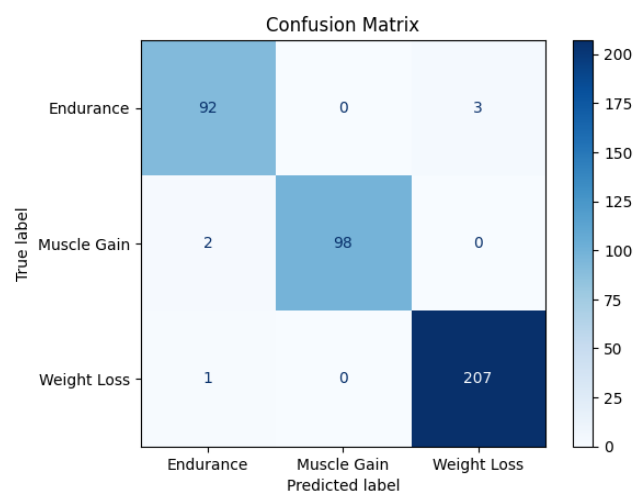
However, in this project, the models were evaluated based on the highest F1-score. This score avoids any issue brought by the unevenly distributed counts of food included in each class, especially on accuracy alone. F1-score is also a function of both precision and recall, which makes it a reliable metric to be used in determining how well the model performs in each class (e.g. how well it avoids false positives and false negatives across all classes).



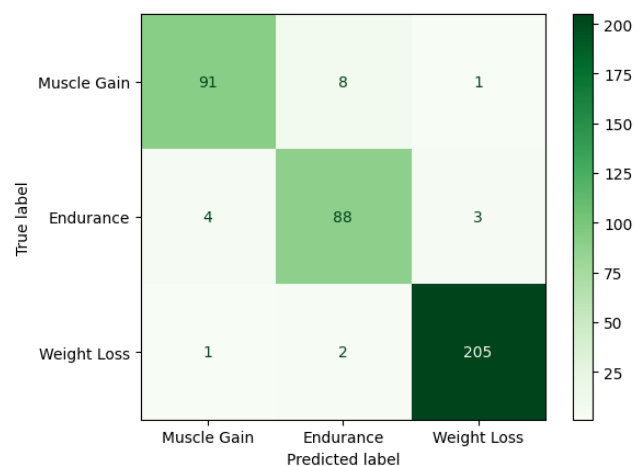
**Figure 1: Confusion Matrix for Naive Bayes**



**Figure 2: Confusion Matrix for MLP**



**Figure 3: Confusion Matrix for AdaBoost**



**Figure 4: Confusion Matrix for SVM**

**Adaboost** achieved the highest F1-score of 98.33%, with only 6 total misclassifications out of 403 predictions. Re-iterating the theory behind Adaboost, it is an ensemble method that combines multiple weak classifiers and adds more weight on the learning of weaker classifiers. For example, if some food were labeled closer to the threshold values, Adaboost could focus on training these values more to improve their classification across each iteration. The simple threshold labeling method, with the clear threshold logic, also strengthened the performance of each of the multiple weak learners. The SAMME algorithm also improved the predictions, helping the model distinguish categories with overlapping nutrient profiles.

**MLP** achieved the next best F1-score at 96.15% which showed how the model was able to learn complex boundaries and overlapping cases. However, it was not able to surpass the F1-score of Adaboost, potentially due to the size of the data set that made it difficult for the model to classify borderline examples.

**SVM** achieved the third best F1-score at 93.69%, showing how the model maximizes the hyperplane margin between different classes. However, overlapping values and near-decision boundaries could have dampened the performance of SVM since the model typically commits to a single decision boundary. Compared to AdaBoost as well, SVM treats all errors equally. Additionally, this could also be computationally heavier than AdaBoost due to the use of non-linear kernel mapping (RBF).

**Naive Bayes** performed the least ideal among all classes at 89.20%. Perhaps, due to the fact that a requirement of the naive assumption requires having the features be conditionally independent. Features like calories, protein, and fat could be correlated. Additionally, the output of this model is probabilistic and not adaptive, so misclassified items and ambiguous cases are highly likely to be misclassified as well.

## 8. Further Data Analysis and Visualization

To better understand the three fitness goal categories that we have, especially on the different nutritional components that separate these three, exploratory analyses were conducted. To note, this is not to build another classifier, but to explain why the models behave the way they do and which nutrients actually drive the classification.

### ANOVA (Analysis of Variance)

A one-way ANOVA was performed across all nutrient features of the dataset to identify the specific nutrients that show statistically significant differences among the three classes with a threshold of  $p < 0.05$ .

From the ANOVA, it showed that there are several nutrients that vary strongly between the three fitness goals, this includes phosphorus, protein, caloric value, carbohydrates, potassium, vitamin B3, magnesium, and nutrition density, among others. This means that the three fitness goal categories vary largely in these nutrients.

Table 5: ANOVA Results

Feature	F_score	p_value
Phosphorus	524.4870	$5.1 \times 10^{-184}$
Protein	515.9771	$1.4 \times 10^{-181}$
Caloric Value	446.8096	$2.9 \times 10^{-161}$
Carbohydrates	401.7120	$1.7 \times 10^{-147}$
Potassium	323.5608	$1.5 \times 10^{-122}$
Vitamin B3	295.9874	$2.2 \times 10^{-113}$
Magnesium	289.7449	$2.7 \times 10^{-111}$
Nutrition Density	272.2297	$2.4 \times 10^{-105}$
Fat	228.0782	$5.39 \times 10^{-90}$
Sugars	205.6756	$5.42 \times 10^{-82}$
Zinc	191.1198	$1.03 \times 10^{-76}$
Saturated Fats	184.3119	$3.2 \times 10^{-74}$
Vitamin B6	163.1443	$2.2 \times 10^{-66}$
Calcium	156.3630	$7.66 \times 10^{-64}$
Monounsaturated Fats	153.2323	$1.15 \times 10^{-62}$
Iron	149.2020	$3.83 \times 10^{-61}$
Water	122.2236	$8.07 \times 10^{-51}$
Vitamin B5	118.8023	$1.71 \times 10^{-49}$
Polyunsaturated Fats	96.19265	$1.27 \times 10^{-40}$
Cholesterol	70.22333	$3.32 \times 10^{-30}$
Dietary Fiber	57.47841	$5.31 \times 10^{-25}$
Vitamin B1	56.89568	$9.22 \times 10^{-25}$
Vitamin E	50.83377	$2.91 \times 10^{-22}$
Vitamin B2	45.29640	$5.73 \times 10^{-20}$
Sodium	15.88552	$1.43 \times 10^{-7}$
Manganese	14.35253	$6.47 \times 10^{-7}$
Vitamin D	9.72519	$6.26 \times 10^{-5}$
Selenium	7.829402	0.00041
Copper	5.403746	0.004565
Vitamin A	1.003575	0.36675
Vitamin B11	0.811763	0.44422
Vitamin C	0.479673	0.619057
Vitamin K	0.332647	0.717063
Vitamin B12	0.240404	0.786332

Significant features ( $p < 0.05$ ):

```
[ 'Phosphorus', 'Protein', 'Caloric_Value',
  'Carbohydrates', 'Potassium', 'Vitamin_B3',
  'Magnesium', 'Nutrition_Density', 'Fat',
  'Sugars', 'Zinc', 'Saturated_Fats', 'Vitamin_B6',
  'Calcium', 'Monounsaturated_Fats', 'Iron',
  'Water', 'Vitamin_B5', 'Polyunsaturated_Fats',
  'Cholesterol', 'Dietary_Fiber', 'Vitamin_B1',
  'Vitamin_E', 'Vitamin_B2', 'Sodium', 'Manganese',
  'Vitamin_D', 'Selenium', 'Copper' ]
```

These nutrients were expected to be the top nutrients with very high F-scores since muscle gain foods are high in protein, and caloric values, endurance foods are high in carbo-

hydrates and sugar, and weight loss foods are low in fat, carbohydrates, and calories. These are just some of the observations that were translated into the assigning of the label code block of the project.

Using ANOVA, it also noted that there are other significant features in each fitness goal category on top of the ones used in our label assignment code block.

#### Extended Feature Columns Included in the Dataset used in the Model

These significant features listed by ANOVA were then tested to be included in the dataset used in the 4 algorithms, with the main question to answer “will the performance of the models differ if we introduce the extended feature columns?”

After doing such, the answer we got is it did not make the performance of the models better, but rather, made it worse. The primary reason here is that the feature space of the model increased but we did not update the code block setting the criteria for the food classification. As such, the additional feature columns became noise to the models as they were never used in label definition.

#### Logistic Regression

Lastly, in this project, logistic regression was also used, not as an additional model to be compared, but as a tool to interpret the results. This is to understand how the nutrients influence the likelihood of that food being labeled into one of the categories.

The regression model was fitted both with all features in the dataset, and with the significant features identified from ANOVA. A positive coefficient means that it increasing that nutrient in the food will make its probability of being assigned to a certain class higher, while a negative coefficient makes it otherwise.

The results of logistic regression fitted using all features are as follows: for muscle gain foods, caloric value, and protein, among other nutrients showed strong positive weights for this class; for endurance foods, carbohydrates and sugars had positive values, and; for weight loss foods, protein, calories, and fat had negative weights. These results support the criteria used in assigning the labels.

For the logistic regression that fitted only the significant features found in the ANOVA above, it showed a somewhat similar set of results to the logistic regression fitted using all features. This reinforced the findings of the earlier logistic regression.

**Table 6: Logistic Regression Results (Significant Features Only)**

Feature	Endurance	Muscle Gain	Weight Loss
Phosphorus	0.060323	1.041003	-1.10133
Protein	-2.58352	7.248356	-4.66484
Caloric Value	0.356608	3.048276	-3.40488
Vitamin B3	-0.07993	0.897341	-0.81741
Magnesium	0.249037	0.075675	-0.32471
Nutrition Density	-0.01557	0.867834	-0.85226
Zinc	-0.71354	0.369557	0.343986
Saturated Fats	0.633249	-0.24054	-0.39271
Vitamin B6	-0.13179	0.060848	0.070945
Iron	0.203868	-0.21558	0.011716
Water	0.101994	-0.0795	-0.0225
Vitamin B5	-0.12971	0.267151	-0.13744
Dietary Fiber	-0.27226	-0.33315	0.605416
Vitamin B1	-0.42197	0.218607	0.203367
Vitamin E	-0.08733	0.095735	-0.00841
Manganese	0.088494	0.090736	-0.17923
Vitamin D	0.11064	-0.32421	0.213571
Selenium	-0.04642	0.052873	-0.00645
Carbo-hydrates	2.478445	0.726385	-3.20483
Potassium	0.121585	0.019321	-0.14091
Copper	-0.07213	0.050346	0.021785
Fat	0.461132	0.50662	-0.96775
Sugars	1.504261	0.670178	-2.17444
Sodium	-0.1375	-0.12352	0.261014
Calcium	-0.08835	-0.59106	0.679415
Mono-unsaturated Fats	0.60894	-0.07883	-0.53011
Vitamin B2	-0.04426	0.03586	0.080121
Poly-unsaturated Fats	0.057274	-0.16954	0.112269
Cholesterol	-0.47535	0.958718	-0.48337

Combining the results of ANOVA and Logistic Regression helped validate the logic used in classifying the entries inside the dataset.

## 5. REFERENCES

- [1] M. Solan, "Why is eating healthy so hard?," Harvard Health, Feb. 01, 2022. <https://www.health.harvard.edu/nutrition/why-is-eating-healthy-so-hard>
- [2] Better Health Channel, "Sporting performance and food," Vic.gov.au, Apr. 17, 2023. <https://www.betterhealth.vic.gov.au/health/healthyliving/sporting-performance-and-food>
- [3] J. Iraki, P. Fitschen, S. Espinar, and E. Helms, "Nutrition Recommendations for Bodybuilders in the Off-Season: A Narrative Review," *Sports*, vol. 7, no. 7, p. 154, Jun. 2019, doi: <https://doi.org/10.3390/sports7070154>.
- [4] E. R. Helms, A. A. Aragon, and P. J. Fitschen, "Evidence-based recommendations for natural bodybuilding contest preparation: nutrition and supplementation," *Journal of the International Society of Sports Nutrition*, vol. 11, no. 1, May 2014, doi: <https://doi.org/10.1186/1550-2783-11-20>.
- [5] "DIETARY CARBOHYDRATE AND THE ENDURANCE ATHLETE: CONTEMPORARY PERSPECTIVES," Gatorade Sports Science Institute, 2022. <https://www.gssiweb.org/research/article/dietary-carbohydrate-and-the-endurance-athlete-contemporary-perspectives>.
- [6] "View of Machine Learning Techniques for Classifying Indonesian Foods and Drinks by Nutritional Profiles," Shm-publisher.com, 2025. <https://www.shmpublisher.com/index.php/joiser/article/view/528/277> (accessed Nov. 17, 2025).
- [7] M. Senthilmurugan, Nagendar Yamsani, C. Mary, Loganayagi S, and A. Akilandeswari, "Evaluation of Support Vector Machine and Kernel Neural Network Classification for Fast Food Nutrition Data," 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), pp. 150–154, Aug. 2023, doi: <https://doi.org/10.1109/ICAISS58487.2023.10250603>.
- [8] A. Ajami and B. Teimourpour, "A Food Recommender System in Academic Environments Based on Machine Learning Models." Accessed: May 24, 2024. [Online]. Available: <https://arxiv.org/pdf/2306.16528>
- [9] A. Ahuja, V. Kedia, M. Ansari, B. Grover, and Shubha Puthran, "Food Classification using Machine Learning Algorithms," *Advances in computer science research*, pp. 1034-1046, Jan. 2025, doi: [https://doi.org/10.2991/978-94-6463-858-5\\_86](https://doi.org/10.2991/978-94-6463-858-5_86).
- [10] Witard, Oliver C et al. "Myofibrillar muscle protein synthesis rates subsequent to a meal in response to increasing doses of whey protein at rest and after resistance exercise." *The American journal of clinical nutrition* vol. 99,1 (2014): 86-95. doi: <https://doi.org/10.3945/ajcn.112.055517>
- [11] Schoenfeld BJ, Aragon AA. How much protein can the body use in a single meal for muscle-building? Implications for daily protein distribution. *J Int Soc Sports Nutr.* 2018 Feb 27;15:10. doi: <https://doi.org/10.1186/s12970-018-0215-1>. PMID: 29497353; PMCID: PMC5828430.
- [12] Slater GJ, Dieter BP, Marsh DJ, Helms ER, Shaw G and Iraki J (2019) Is an Energy Surplus Required to Maximize Skeletal Muscle Hypertrophy Associated With Resistance Training. *Front. Nutr.* 6:131. doi: <https://doi.org/10.3389/fnut.2019.00131>
- [13] Carlsohn A. Recent Nutritional Guidelines for Endurance Athletes. *Dtsch Z Sportmed.* 2016; 67: 7-12. doi: <https://doi.org/10.5960/dzsm.2015.193>
- [14] Naderi, Alireza & Gobbi, Nathan & Ali, Ajmol & Berjisian, Erfan & Hamidvand, Amin & Forbes, Scott & Koozehchian, Majid & Karayigit, Raci & Saunders, Bryan. (2023). Carbohydrates and Endurance Exercise: A Narrative Review of a Food First Approach. *Nutrients*. 15. doi: <https://doi.org/10.3390/nu15061367>
- [15] Julia A Ello-Martin, Liane S Roe, Jenny H Ledikwe, Amanda M Beach, Barbara J Rolls, Dietary energy density in the treatment of obesity: a year-long trial comparing 2 weight-loss diets, *The American Journal of Clinical Nutrition*, Volume 85, Issue 6, 2007, Pages 1465-1477, ISSN 0002-9165, doi: <https://doi.org/10.1093/ajcn/85.6.1465>.
- [16] Clark, Michelle J, and Joanne L Slavin. "The effect of fiber on satiety and food intake: a systematic review." *Journal of the American College of Nutrition* vol. 32,3 (2013): 200-11. doi: <https://doi.org/10.1080/07315724.2013.791194>