CentraleSupélec

2024-2025

Decision Modeling

ELABORATION OF SOME DECISION MODELS FOR THE NUTRI-SCORE LABEL

Jintao Ma

Xianyun Zhuang

Yutao Chen

Supervisor

Prof. Brice MAYAG

Dec 2024

Contents

List of Figures

List of Tables

Abstract

In this report, we implemented a food classification project based on Nutri-Score ratings, following the principles of decision modeling[May25] and machine learning. The project involved multi-criteria decision analysis (MCDA) methods, as well as exploratory data analysis and supervised learning techniques.

(i) In Chapter1, we introduced the Nutri-Score system, highlighting its focus on penalizing unhealthy components like *sugar* and *saturated fats* while rewarding nutrients like *fiber*, *protein*, and *fruits*.

(ii) In Chapter2, we provided an overview of the dataset, including its sources, preprocessing steps such as handling missing values, removing duplicates, and selecting relevant features. This ensured data quality for subsequent analysis.

(iii) In Chapter3and Chapter4,we used the ELECTRE-Tri model with Optimistic and Pessimistic version and two MCDA models combining nutri-score and Eco-score. We established profiles, assigned weights, and thresholds, and compared results between provided and customized weight configurations.

(iv) In Chapter5, we applied machine learning methods. Four models like random forests,K-Nearest Neighbors(KNN),Gaussian Naive Bayes and XGBoost were implemented to predict Nutri-Score categories.Then we also compared results.

(v) In Chapter7, we evaluated and compared the models' performances using key metrics. Additionally, we analyzed feature importance and model-specific insights to draw meaningful conclusions.

At last, we uploaded the code to Github: GitHub Repository.Additionally, we also uploaded our presentation slides to PPT Repository.

1 Introduction

Nutri-Score, a scientific nutrition labeling system, simplifies nutritional information to help consumers make healthier choices in today's fast-paced lifestyle and encourages manufacturers to improve product quality.

The core concept of Nutri-Score is to simplify complex nutritional information into a five-color and letter-coded system,as shown in figure 1.1,ranging from Green A (indicating the highest nutritional value) to Red E (indicating lower nutritional value). This scoring method not only enables consumers to easily understand the health properties of food but also supports the implementation of public health policies. Nutri-Score's calculation method is grounded in scientific research and considers both the components that should be limited and those beneficial to health:

- **Components to avoid** include energy (calories), sugar, saturated fat, and salt, as excessive intake of these can increase the risk of obesity, hypertension, and cardiovascular diseases.
- **Beneficial components** include protein, dietary fiber, and the proportion of healthy foods such as fruits, vegetables, and nuts. These contribute essential nutrients, support bodily functions, and reduce the risk of chronic diseases.

By balancing the effects of these two categories of components, Nutri-Score generates a composite score that reflects the overall nutritional value of the food product. This scoring system has been adopted in several countries and has become a focal point for both consumers and public health authorities.

Figure 1.1: Nutri-score Definition

2 Dataset

The foundation of any analysis lies in the quality and comprehensiveness of the dataset. To ensure the relevance and accuracy of our findings, we developed a carefully curated database, leveraging publicly available resources and implementing robust preprocessing techniques.

2.1 Dataset Overview

The dataset was sourced from the ***Open Food Facts API*** [1] , a widely recognized platform providing extensive food-related data. Given the scope of our study and the need to focus on specific categories, we extracted data from the following product groups:

- Snacks
- Biscuits
- Cereals
- Meals
- Beverages
- Cheeses
- Fruits-based Products

To maintain the integrity of our research and ensure originality, we designed the database to overlap with other groups' datasets by no more than 30%. This precaution not only avoids redundancy but also enables the creation of unique insights within the framework of the project.

[1] `https://world.openfoodfacts.org/cgi/search.pl`

2.2 Data Preprocessing

To ensure the data was clean, consistent, and ready for analysis, we implemented a series of preprocessing steps:

1. Handling Missing Values

   Rows with significant missing data were dropped to avoid introducing bias or inaccuracies into the analysis. This ensured a higher level of data reliability.

2. Removing Duplicates

   Duplicate entries, which could distort statistical insights, were identified and removed. This step helped improve the efficiency and precision of subsequent analyses.

3. Label Selection

   For products with multiple labels (e.g., multiple brands or overlapping categories), a single label was carefully selected based on relevance and consistency.

4. Filtering Relevant Columns

   To streamline the dataset for analysis, only the most pertinent columns were retained. This focused the data on attributes essential for answering research questions while discarding unnecessary information.

5. Random Sampling

   From the filtered dataset, 300 items were randomly sampled to create a manageable yet representative final dataset. This ensured a balance between dataset size and analytical feasibility.

The resulting preprocessed dataset is both tailored to the needs of our study and reflective of real-world products. This database serves as the foundation for deriving meaningful insights shown as 2.1 and 2.2 , enabling us to explore patterns and trends in food categories while maintaining the highest standards of data integrity.

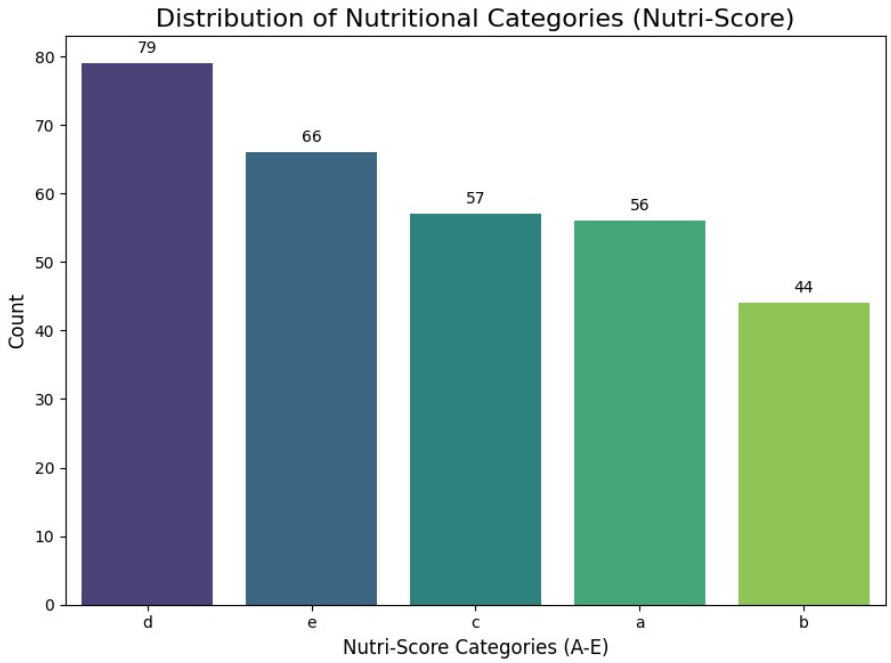

Figure 2.1: distribution of nutrition categories

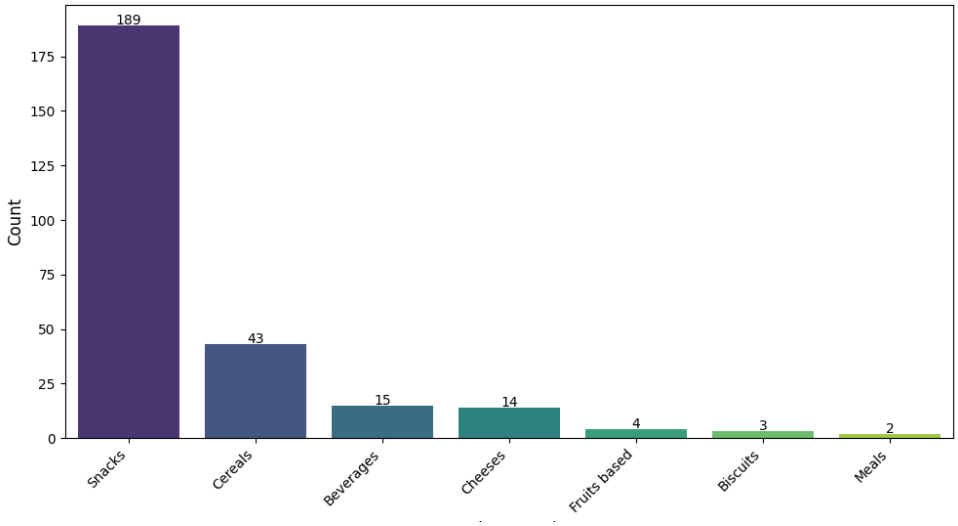

Figure 2.2: distribution of food category

2.3 EDA exploration

To better understand the dataset and extract valuable insights, we conducted a comprehensive exploratory data analysis (EDA) using various visualization techniques and statistical measures. The EDA process involved analyzing individual components, examining relationships between variables, and identifying patterns or anomalies in the data. Below are the key steps and findings:

1. Distribution of Individual Ingredients

Visualization: For each ingredient (e.g., sugar, protein, fiber, salt, saturated fat), bar charts were created to visualize their distribution across the dataset as shown in 2.3.

Analysis: The bar charts revealed the range and frequency of values for each ingredient.

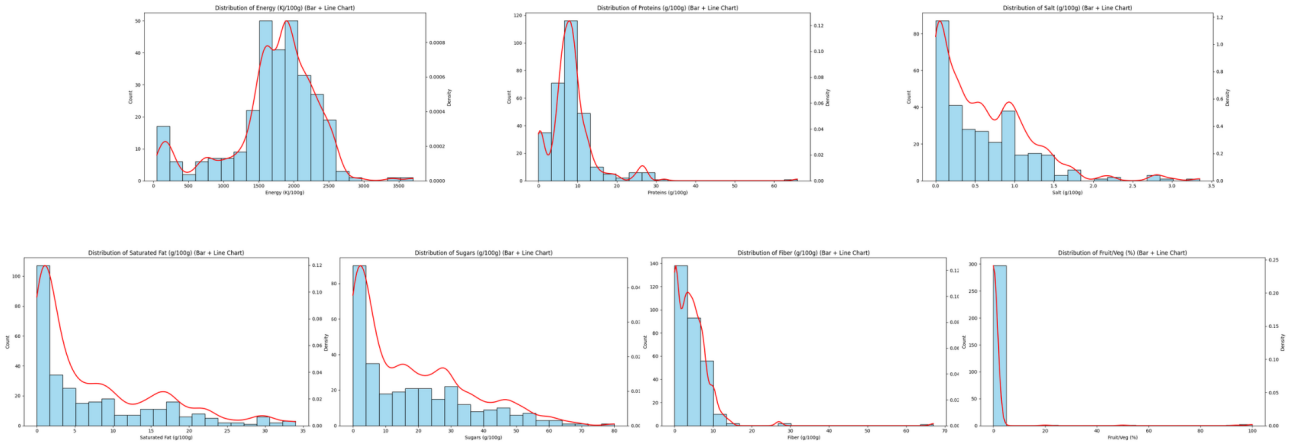

Figure 2.3: bar chart

2. Correlation Analysis Between Ingredients

Visualization: A correlation matrix was generated to examine relationships between every two ingredients as shown in 2.4.

Analysis: Energy is positively correlated with sugars and saturated fat: This suggests that foods high in energy tend to be higher in sugars and saturated fats.

3. Correlation of Energy with other Attributes

There was a correlation between some criteria since the following equation (of the computation of Energy) showing that the criterion “Energy” depended on “Fat,” “Sugars,” “Proteins,” and “Fibers”:

$$\text{Energy} = (9\times\text{fat})+(7\times\text{alcohol})+(4\times\text{protein})+(4\times\text{sugar})+(2.4\times\text{organic acids})+(2.4\times\text{polyols})+(2\times\text{fibers})$$

Our correlation figure 2.5 explained and validated this point.

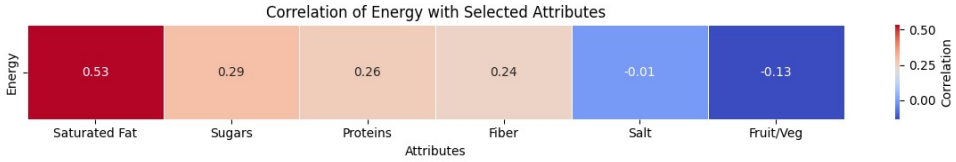

Figure 2.5: Correlation of Energy with Selected Attributes

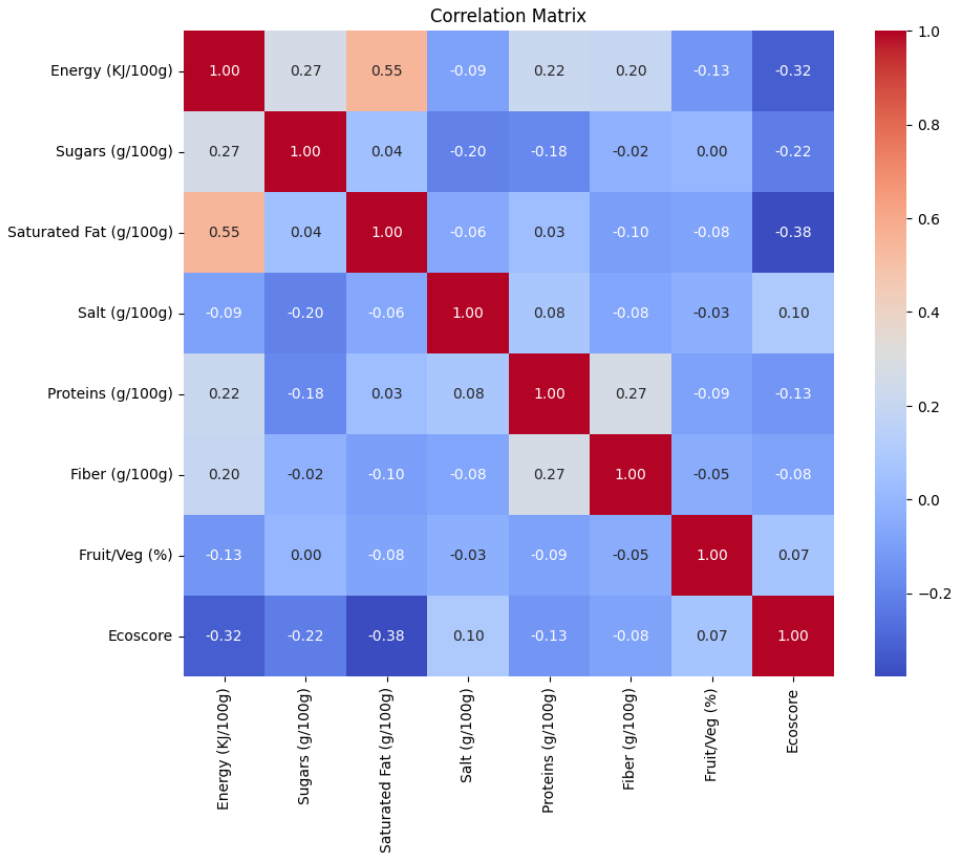

Figure 2.4: correlation matrix

4. Distribution Analysis Using Box Plots

   Visualization: Box plots were created for each ingredient to summarize their distributions, highlighting medians, quartiles, and potential outliers as shown in 2.6 .

   Analysis: Products high in salt and sugar exhibited significant variability, with numerous outliers indicating the diversity in the product categories.

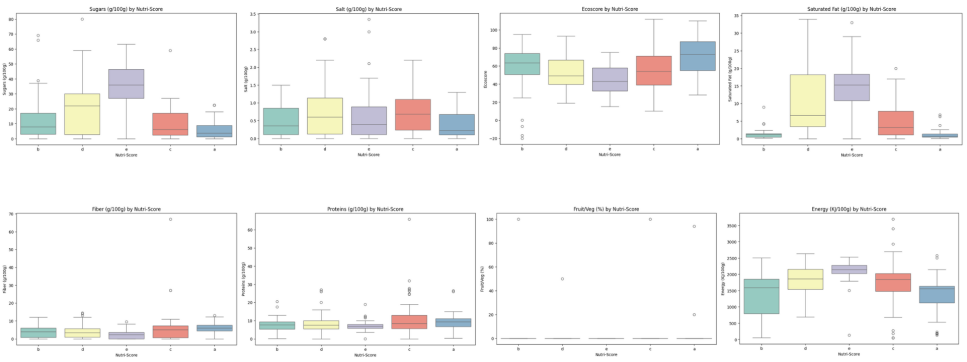

Figure 2.6: Box plots

5. Scatter Plots for Ingredient Relationships

   Visualization: Pairwise scatter plots were generated to visually assess relationships between

every two ingredients (e.g., sugar vs. protein, salt vs. fiber) as shown in 2.7.

Analysis: The scatter plots confirmed trends observed in the correlation matrix. For example, a downward-sloping trend was evident between sugar and protein, reaffirming their negative correlation.

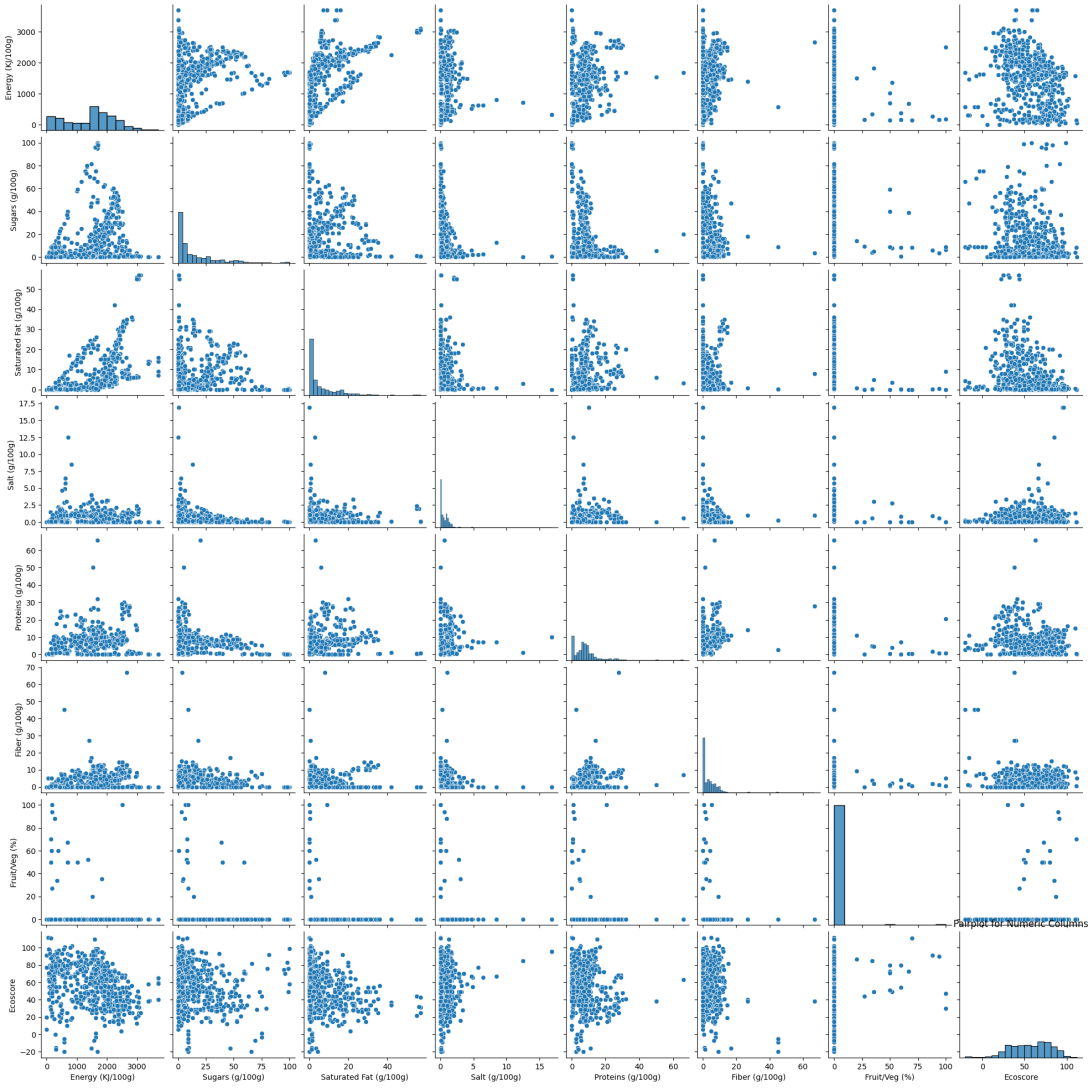

Figure 2.7: Scatter Plots

2.4 Comparison with Other Group

We compared our dataset with database[2] of Beard Guys[3] based on *barcode* column.It showed that overlap percentage was 12.58% and satisfied less than 30% requirement.

[2] https://docs.google.com/spreadsheets/d/16jbL3unDd-vght0ZP012ctZTwaEuLSHX/edit?usp=sharing&ouid=107089449053844369351&rtpof=true&sd=true

[3] Group of BDMA students Muhammad Qasim Khan,Gabriel Lozano Pinzon,Benjamin Gold

3 ELECTRE-TriModel

ELECTRE TRI[Roy02] is a multi-criteria decision analysis (MCDA) method used for sorting or classifying alternatives into predefined categories[CMM12] based on their performance across multiple criteria. Unlike ranking methods, ELECTRE TRI focuses on categorization, leveraging outranking relationships to determine whether one alternative is at least as good as another. It uses thresholds, such as preference, indifference, and veto thresholds, to account for uncertainty and imprecision in decision-making. The method is widely applied in areas like risk assessment, resource allocation, and policy evaluation, where alternatives need to be grouped rather than ranked.

3.1 Establishing Profiles

In constructing profiles for the Electre-Tri model, we used percentile-based divisions named Quintile Method[1] as a fundamental approach. This section described how we created the profiles.

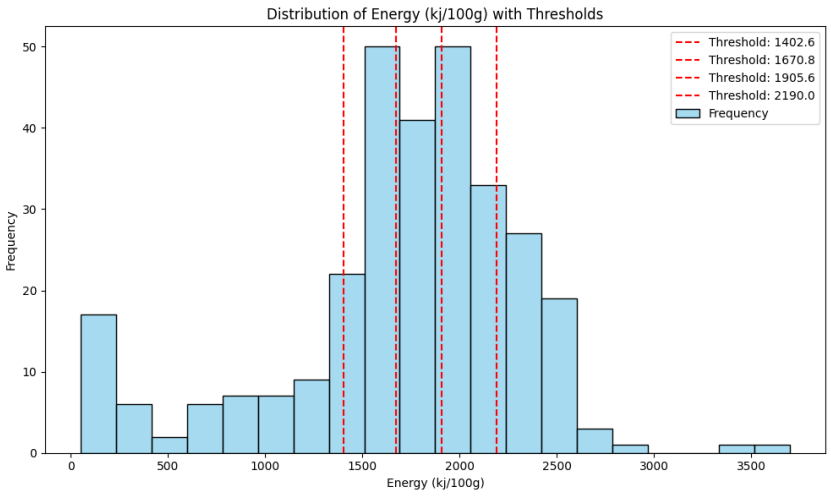

Figure 3.1: Energy profile

[1] https://www.investopedia.com/terms/q/quintile.asp

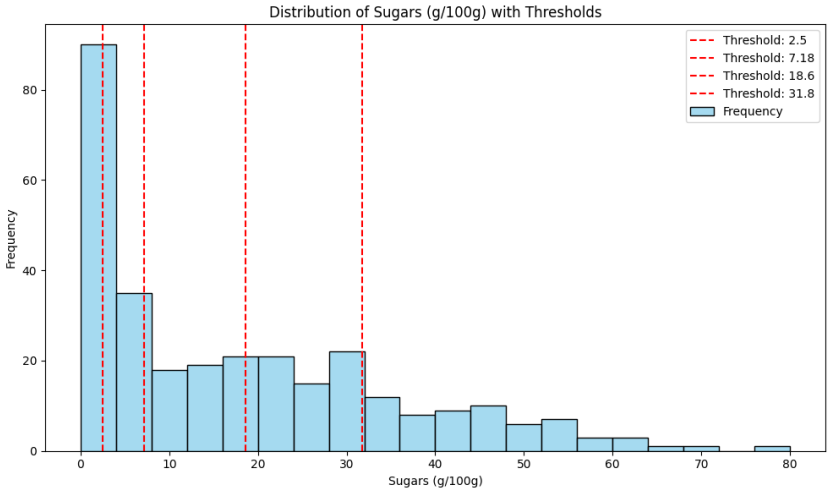

Figure 3.2: Sugar profile

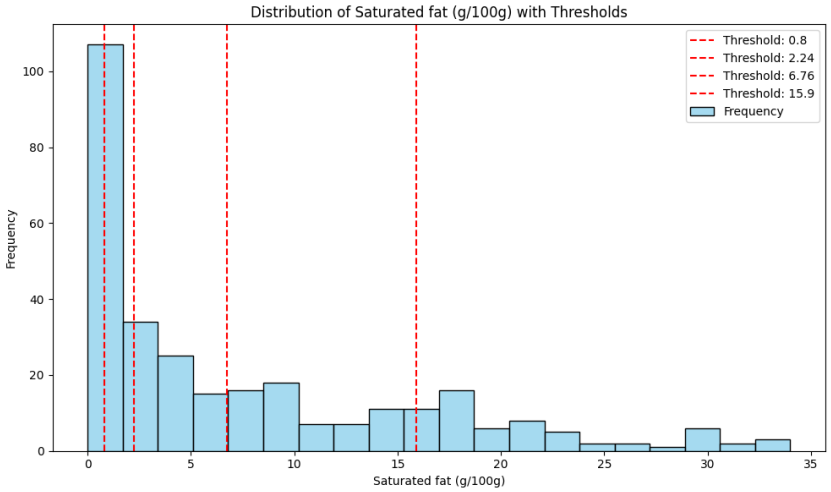

Figure 3.3: Fat profile

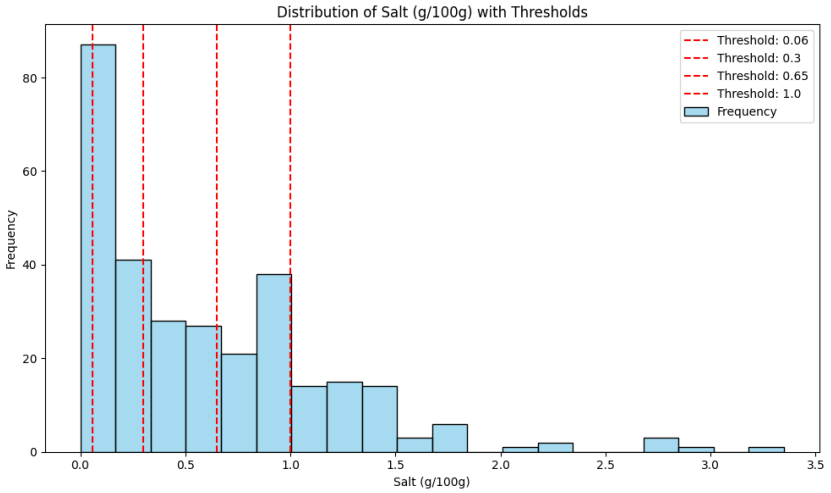

Figure 3.4: Salt profile

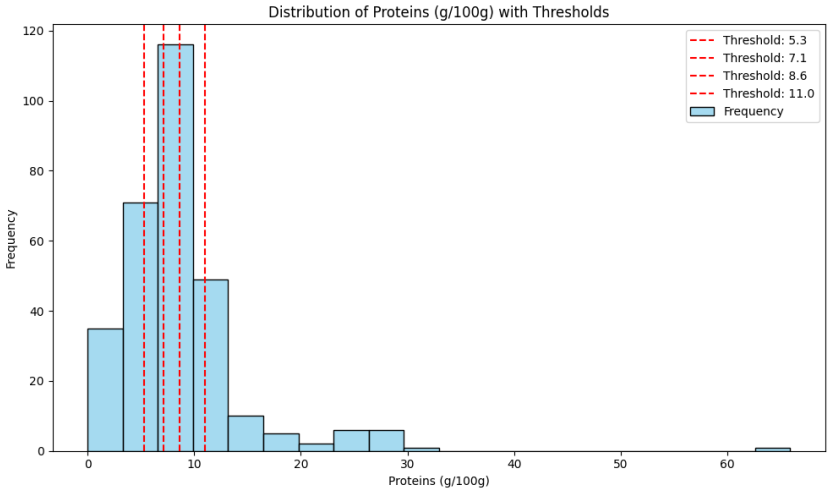

Figure 3.5: Protein profile

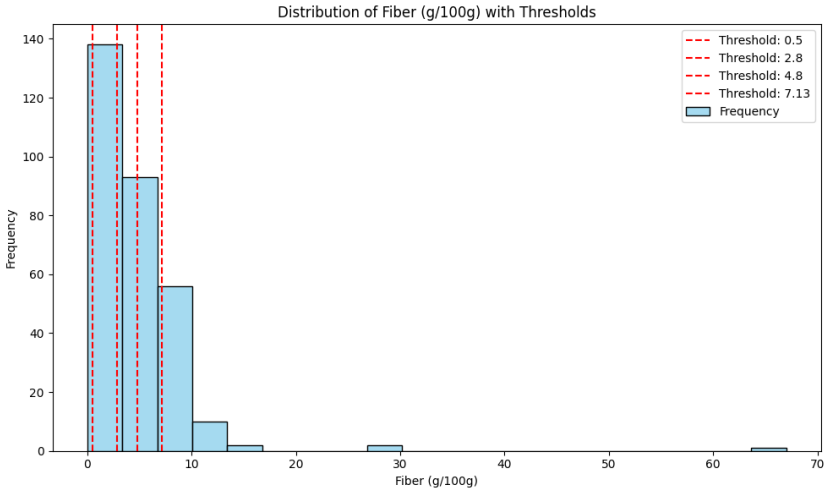

Figure 3.6: Fiber profile

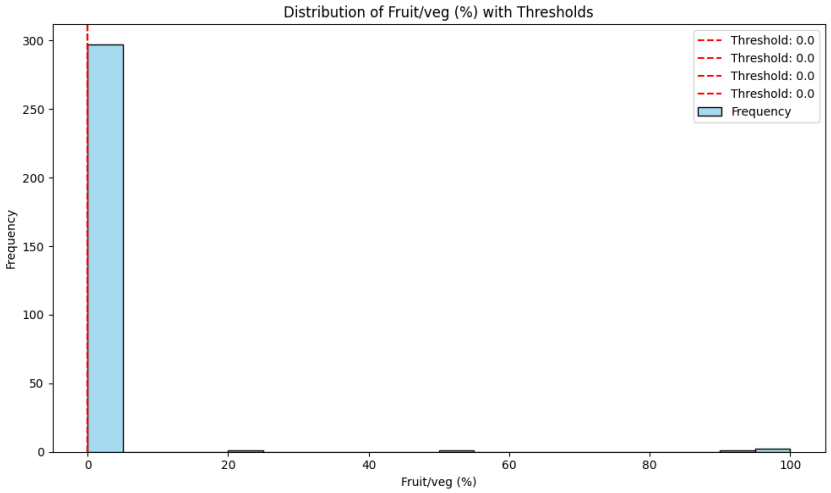

Figure 3.7: Fruits profile

Figures 3.1 to 3.7 visually illustrated the distribution of nutritional values across the dataset, helping to comprehend the spread, central tendencies, and variability of these attributes. The red lines represented the four boundaries.This visualization played a key role in defining the profiles and setting preference thresholds.

Some criteria, such as saturated fat and proteins, displayed right-skewed distributions[2]. This skewness implied that these factors tended to accumulate more values near their lower bounds, leading to a concentration of data points closer to the lower threshold. In contrast, attributes that followed a normal distribution showed a more uniform spread, resulting in profiles clustering around similar lower bounds for these skewed factors.

Profiles were created by dividing the dataset into intervals of 20%, providing a thorough yet manageable method for segmentation. The combined summary of five profiles (a, b, c, d, e) was presented in the table3.1, defining four limits. In the selected categories, the majority were not primarily composed of fruits, and the distribution of fruit content was concentrated around 0% or close to 0%. Therefore, the threshold for fruit content was set to 0.

Profile Limit	**Energy**	**Sugar**	**Fat**	**Salt**	**Proteins**	**Fiber**	**Fruit**
#1	1402.6	2.5	0.8	0.06	11.0	7.13	0.0
#2	1670.8	7.18	2.24	0.3	8.6	4.8	0.0
#3	1905.6	18.6	6.76	0.65	7.1	2.8	0.0
#4	2190.0	31.8	15.9	1.0	5.3	0.5	0.0

Table 3.1: Electre-Tri Model: Profile limits for different nutritional values

Minimization indicators (e.g., Energy, Sugar, Fat, Salt) classified foods from low to high values, where *a* was the best (≤ threshold #1) and *e* was the worst (> threshold #4). Maximization indicators (e.g., Proteins, Fiber) classified foods from high to low values, where *a* was the best (≥ threshold #1) and *e* was the worst (< threshold #4). The table3.2 translated these thresholds into actionable intervals for each category, ensuring accurate classification based on the food's nutritional content.

As table 3.3 shown,we also refer to Minimization indicators as unhealthy factors and Maximization indicators as healthy factors. The direction of the indicator—whether it should be minimized or maximized—determines its classification as a "healthy" or "unhealthy" factor. Minimizing something detrimental and maximizing something beneficial aligns with the our goals for improvement.

[2] `https://www.investopedia.com/terms/s/skewness.asp`

Nutrient	a (Best)	b	c	d	e (Worst)
Energy (KJ/100g)	≤ 1402.6	$(1402.6, 1670.8]$	$(1670.8, 1905.6]$	$(1905.6, 2190.0]$	> 2190.0
Sugars (g/100g)	≤ 2.5	$(2.5, 7.18]$	$(7.18, 18.6]$	$(18.6, 31.8]$	> 31.8
Saturated Fat (g/100g)	≤ 0.8	$(0.8, 2.24]$	$(2.24, 6.76]$	$(6.76, 15.9]$	> 15.9
Salt (g/100g)	≤ 0.06	$(0.06, 0.3]$	$(0.3, 0.65]$	$(0.65, 1.0]$	> 1.0
Proteins (g/100g)	≥ 11.0	$[8.6, 11.0)$	$[7.1, 8.6)$	$[5.3, 7.1)$	< 5.3
Fiber (g/100g)	≥ 7.13	$[4.8, 7.13)$	$[2.8, 4.8)$	$[0.5, 2.8)$	< 0.5
Fruit/Veg (%)	≥ 0.0	≥ 0.0	≥ 0.0	≥ 0.0	< 0.0 (N/A)

Table 3.2: Thresholds and intervals for nutrient classification into categories a, b, c, d, and e.

Factors	Indicators	Nutritional Values
Healthy	Maximization	Proteins, Fiber, Fruit
Unhealthy	Minimization	Energy, Sugar, Fat, Salt

Table 3.3: Classification of Nutritional Factors Based on Indicators

3.2 ELECTRE-Tri Model

3.2.1 Weights and Thresholds

Weights	Energy	Sugars	Fats	Salt	Proteins	Fiber	Fruits
w_i	4	3	3	3	2	2	1

Table 3.4: Electre-Tri Model: Provided weights

Weights	Energy	Sugars	Fats	Salt	Proteins	Fiber	Fruits
w_i	1	1	1	1	4	4	1

Table 3.5: Electre-Tri Model: Our weights

In our evaluation, we will evaluate the models using three distinct thresholds: 50%, 60%, and 70%. These thresholds are calculated by multiplying the threshold value (λ in percentage) with the product of the total number of profiles for the model and the sum of the weights, and then dividing by 100, as shown in Equation3.1.

$$\lambda = \frac{\lambda_{\text{in\%}} \times \text{Total number of profiles} \times \text{Sum of weights}}{100} \tag{3.1}$$

3.2.2 Two Different Models

Label	Pessimistic Version	Optimistic Version
Iteration Direction	From best to worst ("a" → "e")	From worst to best ("e" → "a")
Assignment Logic	Assigns the first category where concordance ≥ λ	Assigns the first failed category where concordance < λ
Stopping Condition	Stops once a category satisfies concordance ≥ λ	Stops once a category fails concordance < λ
Category Assigned	Assigns the highest category that satisfies the condition	Assigns the category just before failing the condition
Sample	If $\lambda = 0.7$, a category is assigned if concordance ≥ 0.7 at the first valid level	If $\lambda = 0.7$, a category is assigned if concordance < 0.7 in the next level

Table 3.6: Pessimistic vs. Optimistic Version Assignment Logic

Here we used two functions for Pessimism and Optimism methods. And we listed the table3.6 to compare them.

Pessimism

The pessimistic majority sorting function evaluated each item from the worst category (e) to the best category (a). It assigned a category to each row (item) in the dataset based on whether its concordance ratio with the threshold profile of a category met or exceeded a given lambda threshold (lambda_threshold). The process stopped once the item was classified as meeting the conditions for a particular category. This approach was conservative in nature because it assumed that an item belonged to the worst category by default and only upgraded its classification when it confidently met the conditions of a better category.

Optimism

The optimistic majority sorting function worked in the opposite direction, starting from the best category (a) and moving towards the worst category (e). For each item, it checked whether the concordance ratio with the current category's threshold profile was below the lambda threshold (lambda_threshold). If the item failed to meet the conditions for a particular category, it was classified into the previous (worse) category. This method was optimistic because it assumed that the

item belonged to the best category by default and only downgraded its classification when the conditions for the higher category were not met.

3.3 Results Analytics

In this project, we will evaluate two Electre-Tri models: the optimistic and the pessimistic versions. The evaluation process will involve testing different combinations of weights and thresholds, as described in previous sections. As a result, the assessment will consist of a total of 12 distinct evaluations (2 models × 2 weight sets × 3 threshold levels).

3.3.1 Provided weight

First we modeled the classification with the given weights (4 3 3 3 2 2 1).

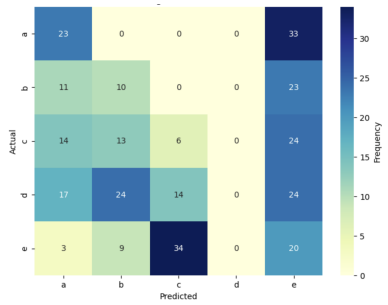

Figure 3.8: $\lambda = 5$

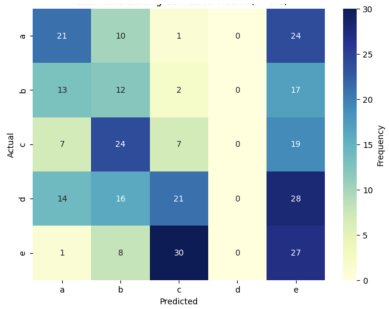

Figure 3.9: $\lambda = 6$

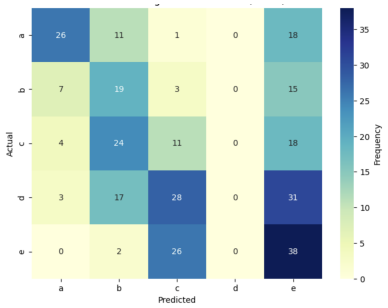

Figure 3.10: $\lambda = 7$

Figure 3.11: Pessimistic Results

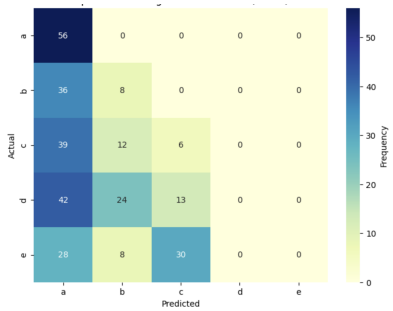

Figure 3.12: $\lambda = 5$

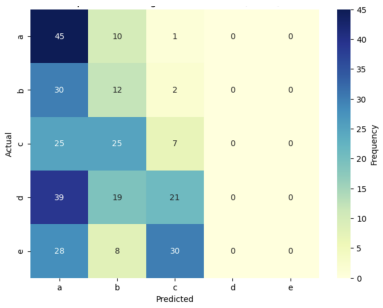

Figure 3.13: $\lambda = 6$

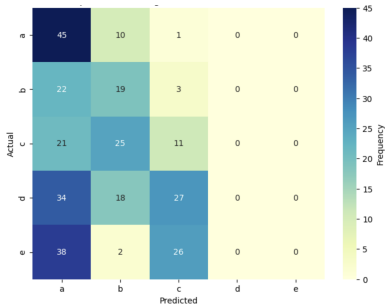

Figure 3.14: $\lambda = 7$

Figure 3.15: Optimistic Results

Here the horizontal axis of the picture was the predicted categorization and the vertical axis direction was the actual categorization. We had two important findings:

the Value of Thresholds

When the λ value increased, this meant the higher the classification threshold, the higher the probability of classification to *d* or *e*; when the λ value was smaller, this meant the lower the classification threshold, the higher the probability of classification to *a* or *b*.

Pessimistic or Optimistic

When we compared the pessimistic and optimistic classification models, we also found that the optimistic classification model was more likely to classify to a higher category such as *a* or *b*, while the pessimistic classification model was more likely to classify to a lower category such as *d* or *e*.

3.3.2 Our Weight

Secondly we modeled the classification with the given weights (1 1 1 1 4 4 1).

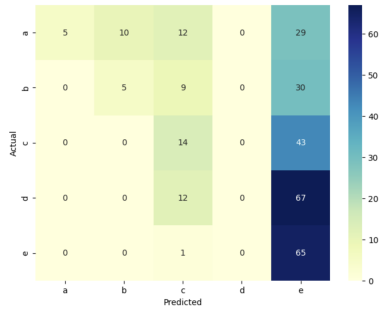

Figure 3.16: $\lambda = 5$

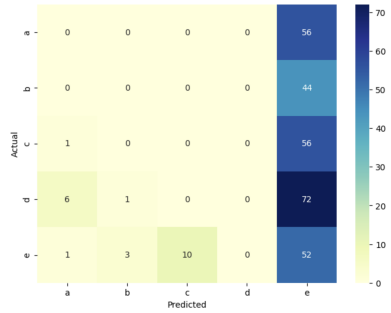

Figure 3.17: $\lambda = 6$

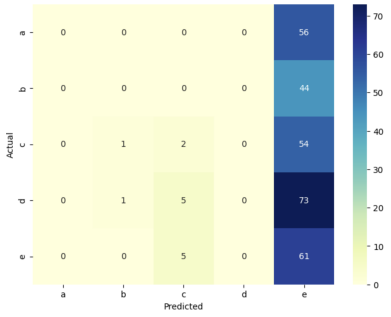

Figure 3.18: $\lambda = 7$

Figure 3.19: Pessimistic Results

Here the horizontal axis of the picture was the predicted categorization and the vertical axis direction was the actual categorization. We had two important findings:

the Value of Thresholds

When the λ value increased, this meant the higher the classification threshold, the higher the probability of classification to *d* or *e*; when the λ value was smaller, this meant the lower the classification threshold, the higher the probability of classification to *a* or *b*.

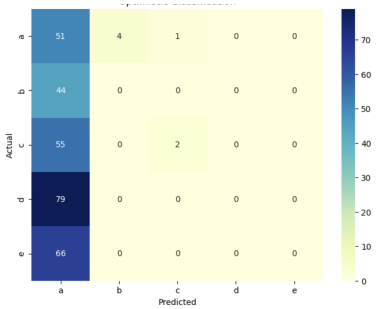

Figure 3.20: $\lambda = 5$

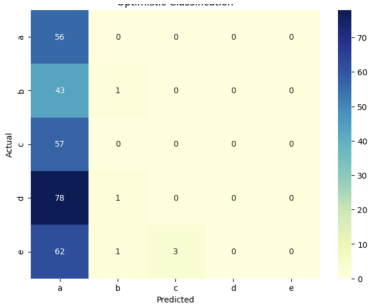

Figure 3.21: $\lambda = 6$

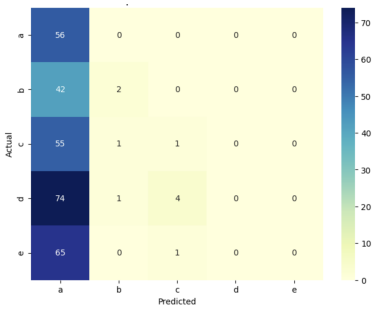

Figure 3.22: $\lambda = 7$

Figure 3.23: Optimistic Results

Pessimistic or Optimistic

When we compared the pessimistic and optimistic classification models, we also found that the optimistic classification model was more likely to classify to a higher category such as *a* or *b*, while the pessimistic classification model was more likely to classify to a lower category such as *d* or *e*.

Impact of different weights of healthy/unhealthy factors

When we decreased the weights of the unhealthy factors and increased the weights of the healthy factors as shown in section3.1, we found that the number of “A” and “B” categories had increased, especially in the optimistic scenario.

That was because pessimistic classification strictly checked categories from highest to lowest, with high weights on healthy factors causing foods to be assigned to lower categories if they failed to meet healthy criteria. It would make results more extreme.

At the same time, the comparison between the results of the pessimistic3.19 and optimistic models3.23 under the second weight was even more contrasting compared to the classification results under the initial weight3.153.15.This was reflected in the more extreme distributions under the second weight under the pessimistic and optimistic models.

3.3.3 Metrics Comparison

For the *weight*(4,3,3,3,2,2,1),we also got 4 important metrics such as:***Accuracy,Precision,Recall,F1-Score*** as figures3.24,3.25,3.26 shown below. Then we also did comparison with machine learning results as figure 5.6,5.6 shown at the Chapter5.

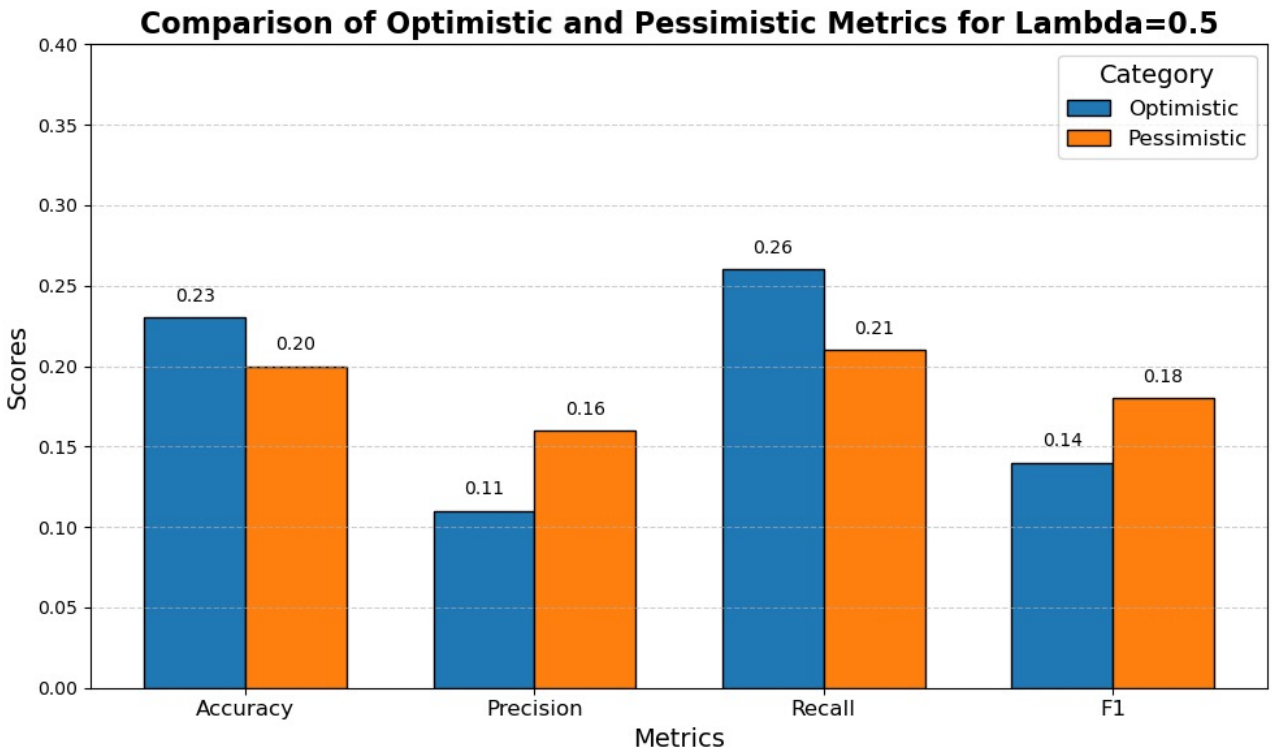

Figure 3.24: Comparison of Optimistic and Pessimistic Metrics for lambda=0.5

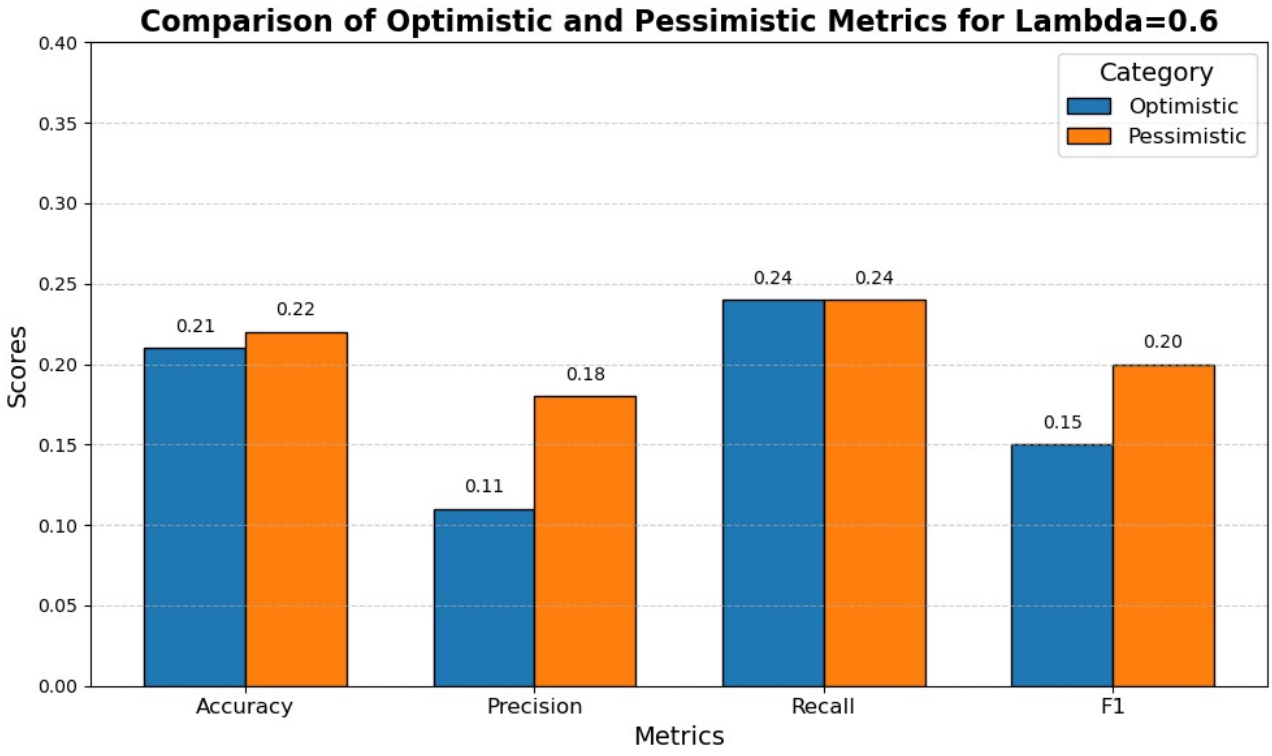

Figure 3.25: Comparison of Optimistic and Pessimistic Metrics for lambda=0.6

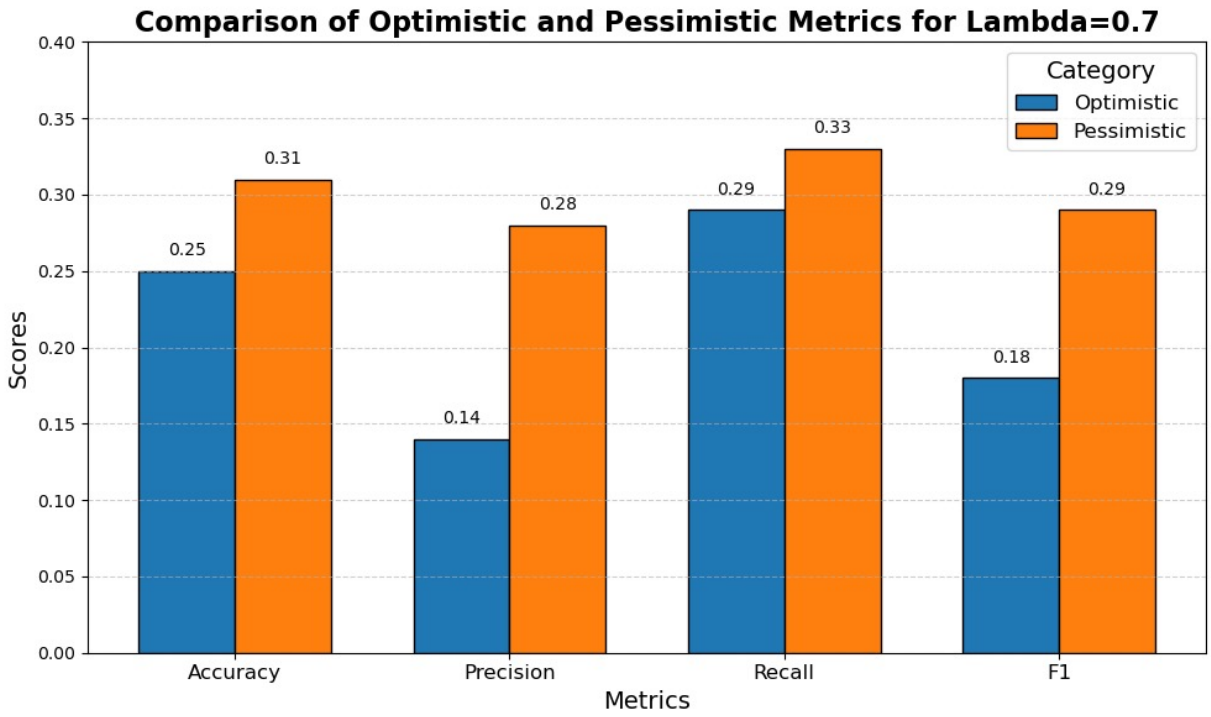

Figure 3.26: Comparison of Optimistic and Pessimistic Metrics for lambda=0.7

Overall Trends:

As λ increased, the Pessimistic model demonstrated significant improvements across all metrics, reflecting the benefits of stricter thresholds in reducing false positives and improving consistency. The Optimistic model, on the other hand, showed limited improvement due to its inherently lenient classification strategy, which resulted in a weaker balance between precision and recall.

Impact of Lambda Threshold:

Higher λ thresholds resulted in stricter categorization, leading to better performance for both models. However, the trade-off was more pronounced for the Optimistic model, which sacrificed recall for stricter precision.

Pessimistic vs. Optimistic:

The Pessimistic model consistently outperformed the Optimistic model across all λ values, particularly at higher thresholds. This advantage stemmed from its conservative approach, which prioritized precision and overall classification accuracy. From the figure 4.1, it can be observed that our data is inherently imbalanced. Specifically, lower-tier categories (such as "D" or "E") have a larger number of samples. The Pessimistic model is likely to focus more on correctly identifying the majority classes, thereby improving overall classification accuracy.

4 MCDA Models

4.1 Distributions of two Scores

From figure4.1 we showed two distributions of $Nutri-score$ and $Eco-score$. We can oberserve that Nutri-score was from e to a while eco-score was from 0 to 100. Then we just fixed threshold values to assign categories (20, 40, 60, 80) for Ecoscore to make it from e to a like table 4.1.

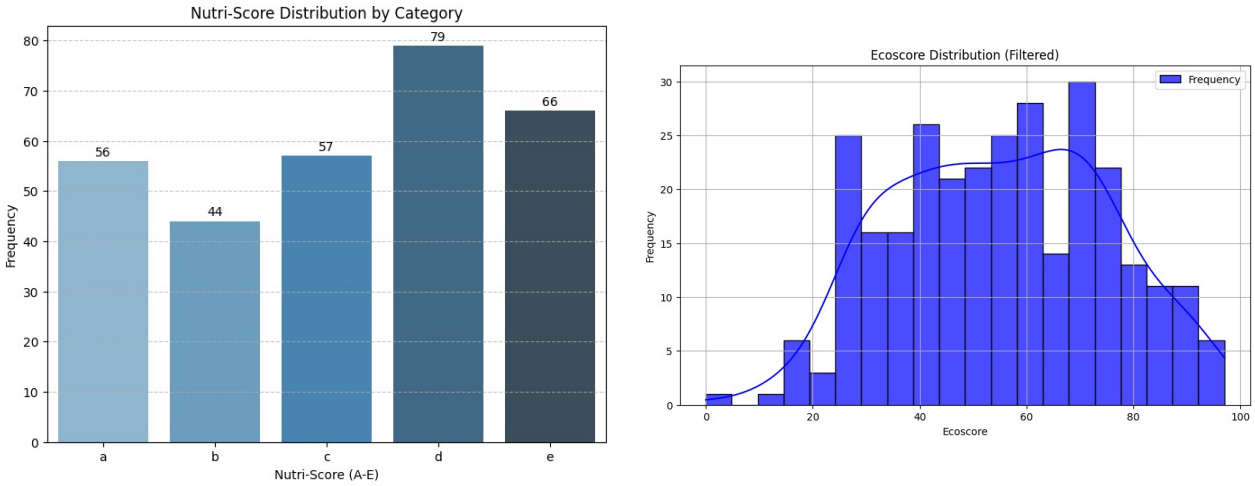

Figure 4.1: Distributions of two Scores

Intervals	a (Best)	b	c	d	e (Worst)
Ecoscore	[80,100)	[60,80)	[40,60)	[20,40)	[0,20)

Table 4.1: Thresholds and intervals for Ecoscore classification into categories a, b, c, d, e

4.2 Weighted Sum Model

The Weighted Sum Model is a widely used Multi-Criteria Decision Analysis (MCDA) approach[BS02] that aggregates multiple criteria into a single evaluation score by normalizing and weighting the criteria. The model operates as follows:

4.2.1 Normalization Using Min-Max Scaling

To ensure comparability between criteria with different ranges, Min-Max Normalization[1]. is applied. The normalized value x' of a criterion x is calculated as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.1}$$

Normalization for Nutri-Score Given Nutri-Score values $a = 5, b = 4, c = 3, d = 2, e = 1$, the normalization formula for a specific Nutri-Score is:

$$N_1 = \frac{\text{Nutri-Score} - 1}{5 - 1} \tag{4.2}$$

Normalization for Eco-Score Assuming that the Eco-Score ranges from 0 to 100, the normalized value is computed as:

$$E_1 = \frac{\text{Eco-Score} - 0}{100 - 0} \tag{4.3}$$

4.2.2 Weight Assignment

Weights are assigned to the normalized criteria to reflect their relative importance. In this model:

$$w_1 = 50\%, \quad w_2 = 50\% \tag{4.4}$$

4.2.3 Aggregation of Scores

The weighted sum of the normalized scores is computed as:

$$\text{Score} = w_1 \cdot N_1 + w_2 \cdot E_1 \tag{4.5}$$

4.2.4 Sorting and Categorization

The aggregated scores are categorized into five groups for ranking purposes:

- **A**: $0.8 \leq \text{Score} \leq 1.0$
- **B**: $0.6 \leq \text{Score} < 0.8$
- **C**: $0.4 \leq \text{Score} < 0.6$
- **D**: $0.2 \leq \text{Score} < 0.4$

[1] https://www.geeksforgeeks.org/data-normalization-in-data-mining/

- **E**: $0.0 \leq \text{Score} < 0.2$

4.2.5 Result Analysis

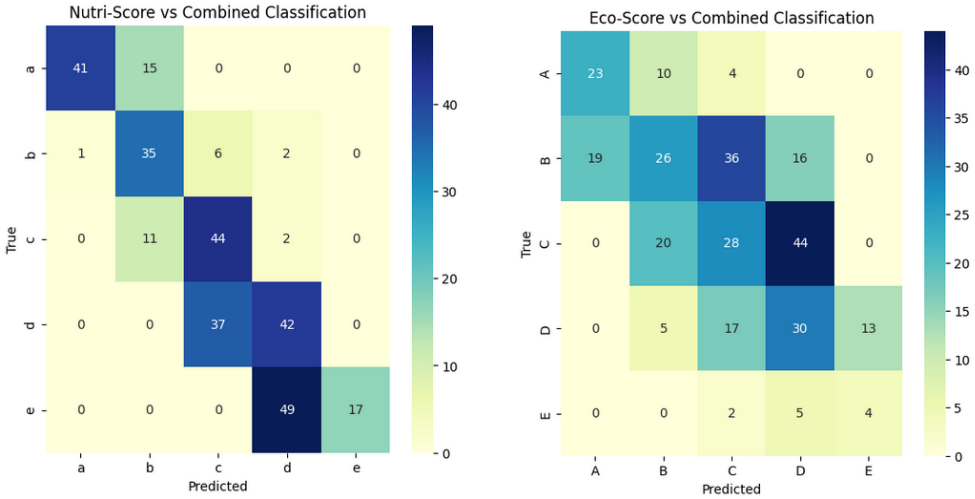

Figure 4.2: Sorting Reults Comparison

The classification results from the Weighted Sum Model are compared with the original classifications based on Nutri-Score and Eco-Score. The comparison is shown in two heat maps Fig: 4.2:

1. **Nutri-Score vs Combined Classification:** The heatmap on the left compares the original Nutri-Score classifications with the combined Weighted Sum Model classifications. The rows represent the true classes (a, b, c, d, e), and the columns represent the predicted classes. The diagonal elements indicate correct classifications, while off-diagonal elements represent misclassifications.

2. **Eco-Score vs Combined Classification:** The heatmap on the right shows the comparison between the original Eco-Score classifications and the combined Weighted Sum Model classifications. Similarly, the rows represent the true classes (A, B, C, D, E), and the columns represent the predicted classes.

The results of the weighted sum classification method are basically consistent with to both of the original Nutri-Score and Eco-Score classification results.

4.3 Simple Decision Rules Model

The Simple Decision Rules Model is an intuitive approach to classification that prioritizes one dimension as the primary criterion (Nutri-Score) while using a secondary dimension (Eco-Score)

to make adjustments when necessary. The rules are structured as follows:

4.3.1 Decision Rules

The rules combine Nutri-Score and Eco-Score classifications into a unified sorting rule, as summarized in Table 4.2.

Nutri-Score	Eco-Score	Sorting-Rules
A	A	A
A	B/C	A
A	D/E	B
B	A/B	B
B	C/D/E	C
C	A/B/C	C
C	D/E	D
D	A/B	C
D	C/D/E	D
E	A/B	D
E	C/D/E	E

Table 4.2: Decision rules combining Nutri-Score and Eco-Score classifications.

4.3.2 Sorting Results Comparison

The performance of the Simple Decision Rules Model is compared to the original classifications using confusion matrices, as shown in Figure 4.3.

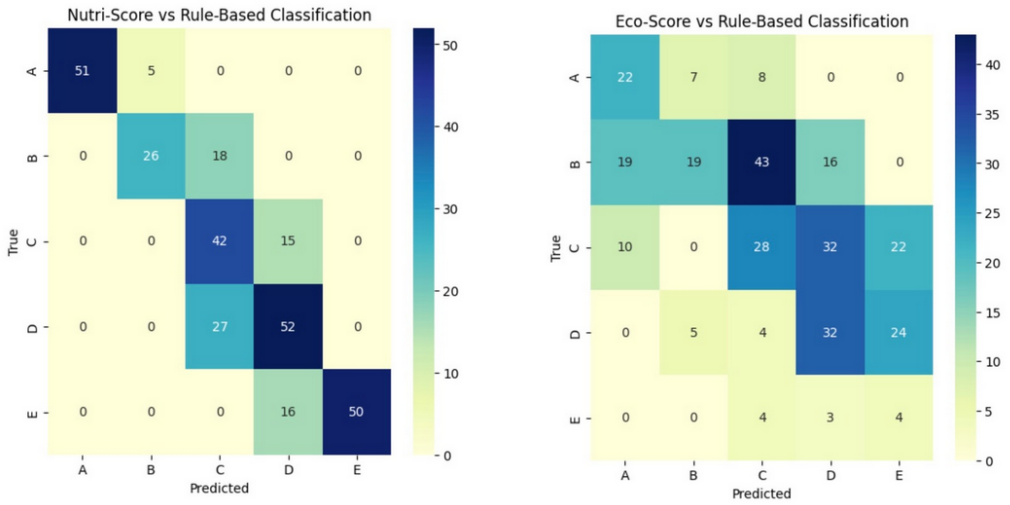

Figure 4.3: Comparison of Sorting Results. **Left:** Nutri-Score vs Rule-Based Classification. **Right:** Eco-Score vs Rule-Based Classification.

4.3.3 Analysis of Results

- **Consistency with Nutri-Score:** The rule-based model demonstrates a high degree of consistency with the original Nutri-Score classifications, as evident in the strong diagonal elements in the left heatmap. This is because the Nutri-Score serves as the primary dimension in the decision-making process.
- **Adjustments Using Eco-Score:** The Eco-Score serves as a secondary dimension for adjustments. This is reflected in off-diagonal elements where the Eco-Score influences the final classification.
- **Improved Robustness:** By prioritizing Nutri-Score while allowing Eco-Score to provide refinements, the model balances simplicity and flexibility, resulting in robust classifications.
- **Misclassifications:** While fewer than in the Weighted Sum Model, some misclassifications persist, particularly where Nutri-Score and Eco-Score suggest conflicting classes. These cases highlight opportunities for refining decision rules.

4.3.4 Conclusion

The Simple Decision Rules Model effectively combines Nutri-Score and Eco-Score dimensions to achieve a classification system that is highly consistent with the original Nutri-Score while incorporating secondary refinements from Eco-Score. This approach provides an interpretable and practical method for multi-criteria classification.

4.4 Findings

This section summarizes the key observations and insights derived from the analysis of the Weighted Sum Model and the Simple Decision Rules Model.

4.4.1 Weighted Sum Model Findings

1. **The weight distribution reflects the characteristics of the original metrics:** The weighted sum method assigns balanced weights to Nutri-Score and Eco-Score. If these original metrics effectively reflect the quality of food products, the weighted sum naturally reinforces their impact, resulting in similar classification outcomes.

2. **Normalization reduces the impact of scale differences:** Nutri-Score and Eco-Score are normalized before summation, ensuring their contributions are proportional and unaffected by scale differences. If the normalized scores correlate strongly with the original classifications, the resulting classifications remain consistent.

3. **Threshold-based classification ensures consistency:** The classification thresholds for Nutri-Score and Eco-Score are carefully designed. If these thresholds align well with the distribution of the combined scores, the classifications derived from the weighted sum will naturally align with the original classifications.

4.4.2 Simple Decision Rules Model Findings

1. **Dominance of Nutri-Score in Rule-Based Logic:**

    - The simple decision rule model relies heavily on the Nutri-Score as the primary criterion for categorization.
    - In the rules, the Nutri-Score dictates the baseline category.
    - Since the Nutri-Score determines the primary structure, the resulting classification aligns closely with it.

2. **Limited Influence of Eco-Score:**

    - The Eco-Score acts as a secondary criterion in the decision rules, allowing only slight modifications to the category derived from the Nutri-Score.
    - This limited influence prevents drastic changes in the classification, resulting in a higher similarity to the Nutri-Score.

3. **Aligned Categorization Frameworks:**

    - Both Nutri-Score and Eco-Score follow a similar A-E classification system, where "A" is the best and "E" is the worst.
    - The decision rules are designed to preserve this alignment, naturally favoring outcomes that reflect the Nutri-Score.

5 Machine Learning Models

5.1 Preparation Work

The dataset was split into training and testing sets using an 80:20 ratio[1]:

$$\text{Train Data Size} = 80\% \times N$$

$$\text{Test Data Size} = 20\% \times N$$

where N is the total number of samples in the dataset.
Key challenges identified during splitting:

- Class imbalance: Uneven distribution of classes affects performance metrics.
- Lack of sufficient data: Small dataset size leads to overfitting in some models.

The following models[Bis06] were chosen:

1. Random Forest
2. K-Nearest Neighbors (KNN)
3. Gaussian Naive Bayes
4. XGBoost

[1] `https://www.askpython.com/python/examples/split-data-training-and-testing-set`

5.2 Parameters Choices

1. **Random Forest:**

$$\text{Number of Trees} = 100$$

$$\text{Max Depth} = \text{Auto-determined.}$$

2. **KNN:**

$$\text{Number of Neighbors} = 5$$

3. **Gaussian Naive Bayes:**

$$\text{Gaussian Assumptions:Features assumed to be independent.}$$

4. **XGBoost:**

$$\text{Boosting Type} = \text{Gradient Boosted Trees.}$$

5.3 Model Processing

Each model's setup and confusion matrix analysis is provided:

1. **Random Forest:**

   - **Setup:** Max depth was automatically determined.
   - **Insights:**

$$\text{Accuracy} = 68.99\%$$

$$\text{Bias} \rightarrow \text{Minor bias toward similar classes.}$$

   - Predictions were balanced across all categories.

   Analysis: The confusion matrix shows that the model performs well across most classes. However, the model get confuse between class b and c, and class d and e.

2. **K-Nearest Neighbors (KNN):**

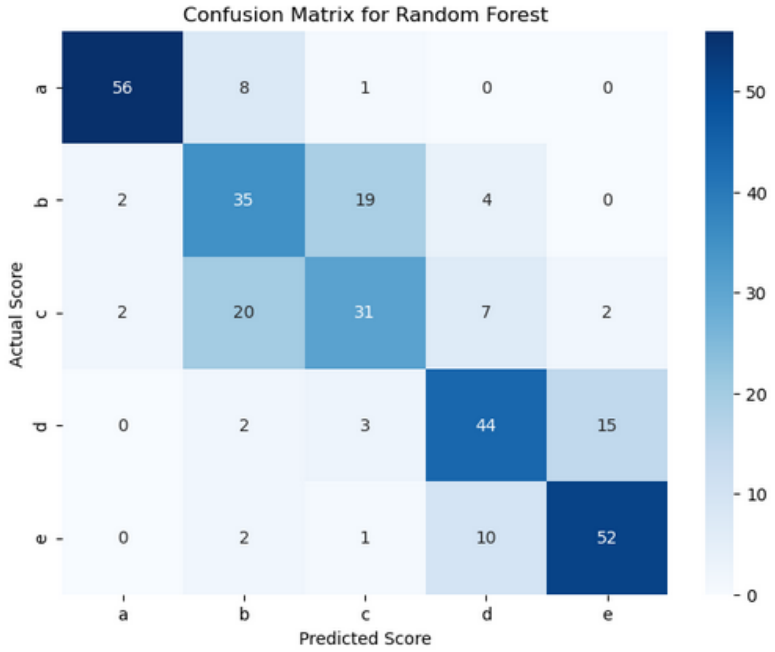

Figure 5.1: Confusion Matrix for Random Forest.

- **Setup:** Number of neighbors = 5.
- **Insights:**

Accuracy = 60.13%

Confusion → Difficulty with neighboring classes.

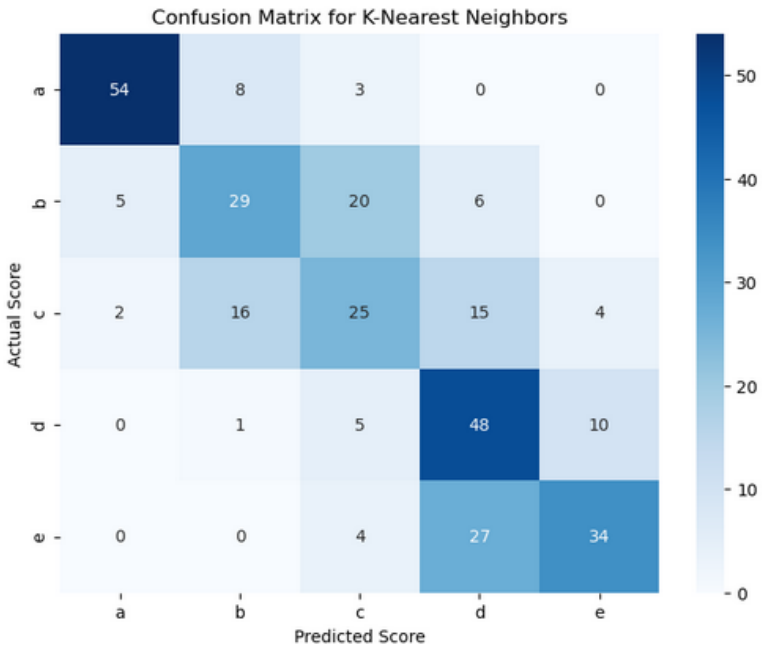

Figure 5.2: Confusion Matrix for K-Nearest Neighbors.

Analysis: The confusion matrix highlights confusion between adjacent classes, particularly

b and *c*. This is expected due to the proximity-based nature of KNN, which struggles with boundary regions.

3. **Gaussian Naive Bayes:**

   - **Setup:** Based on Gaussian distribution assumptions.
   - **Insights:**

$$\text{Accuracy} = 33.23\%$$

$$\text{Limitations} \rightarrow \text{Assumes features are independent.}$$

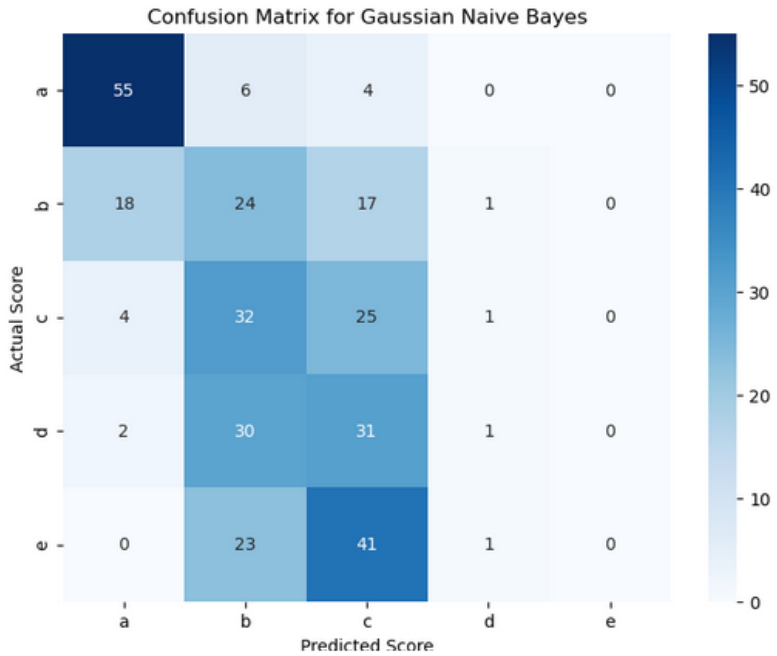

Figure 5.3: Confusion Matrix for Gaussian Naive Bayes.

Analysis: The confusion matrix shows significant misclassifications, particularly for classes *c* and *e*. The independence assumption does not hold for this dataset, leading to poor performance.

4. **XGBoost:**

   - **Setup:** Booster used: GBTrees.
   - **Insights:**

$$\text{Accuracy} = 62.03\%$$

$$\text{Performance} \rightarrow \text{Better recall for intermediate classes.}$$

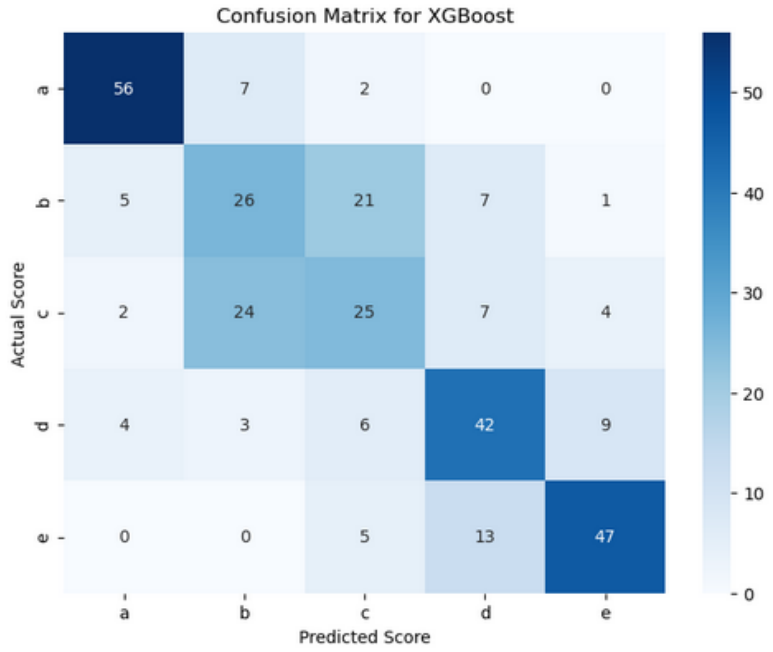

Figure 5.4: Confusion Matrix for XGBoost.

Analysis: The confusion matrix indicates balanced performance across all classes, with better recall for intermediate classes like d. Some confusion remains between b, c, and e, but overall performance is consistent.

5.4 Machine Learning Algorithms Overall Comparison

To evaluate the overall performance of the models, key metrics such as accuracy, precision, recall, and F1-score were compared. The results are summarized in the figure below:

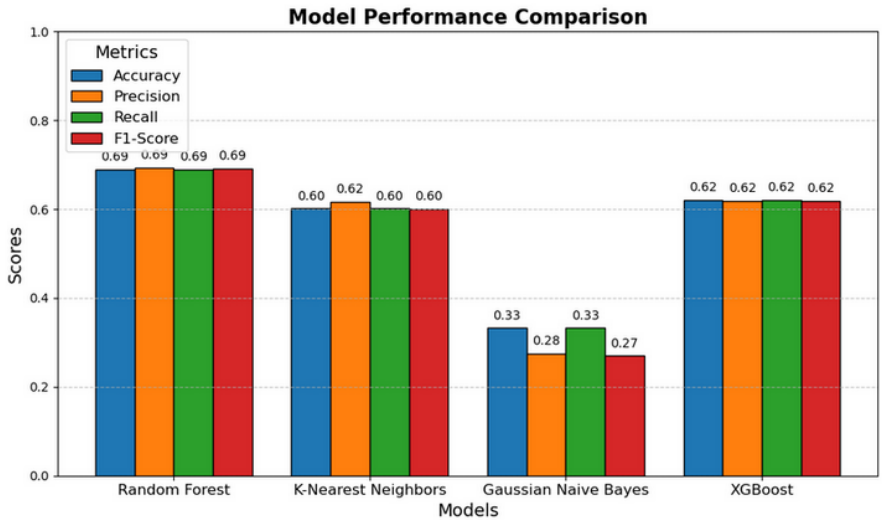

Figure 5.5: Model Performance Comparison Across Metrics.

Insights:

- **Random Forest:** Achieved the highest accuracy (68.99%) among all models, with balanced performance across precision, recall, and F1-score. This model is well-suited for datasets with minor class imbalances.
- **K-Nearest Neighbors (KNN):** Delivered moderate performance (60.13% accuracy). The F1-score matches the precision and recall, indicating consistent, albeit slightly lower, predictive capabilities. Struggles with neighboring class boundaries.
- **Gaussian Naive Bayes:** Demonstrated the lowest performance, with accuracy at 33.23%. The precision, recall, and F1-score are significantly affected by the incorrect assumption of feature independence, as well as sensitivity to class imbalance.
- **XGBoost:** Achieved 62.03% accuracy, slightly outperforming KNN. Shows good recall for intermediate classes and demonstrates consistent precision and recall. However, it underperforms compared to Random Forest for this specific dataset.

Key Observations:

1. **Class Imbalance:** Uneven distributions of classes negatively impacted Gaussian Naive Bayes and KNN the most. Random Forest and XGBoost demonstrated better resilience to imbalance.
2. **Lack of Data:** All models could benefit significantly from a larger, unbiased dataset to improve both true positive rates and reduce false negatives.

5.5 Comparison Between ELECTRE-Tri Model and Machine Learning Models

In this section we provided a comparative analysis of the Pessimistic and Optimistic ELECTRE-Tri models and various machine learning models (Random Forest, K-Nearest Neighbors, Gaussian Naive Bayes, and XGBoost). Table **??** summarizes the performance metrics (Accuracy, Precision, Recall, and F1-score) for both Pessimistic and Optimistic approaches under different lambda thresholds. Figure 5.6, 5.7 visualizes the performance metrics for the Pessimistic ELECTRE-Tri model only, due to space limitations.
Here we used ***Macro Avg***[2] because we wanted to evaluate the model's performance equally across all classes, regardless of class imbalance or the number of samples in each class.

[2] `https://sklearn-evaluation.ploomber.io/en/latest/classification/micro_macro.html`

5.5.1 Key Observations

The results in Figure 5.6, 5.7 highlight the following key findings:

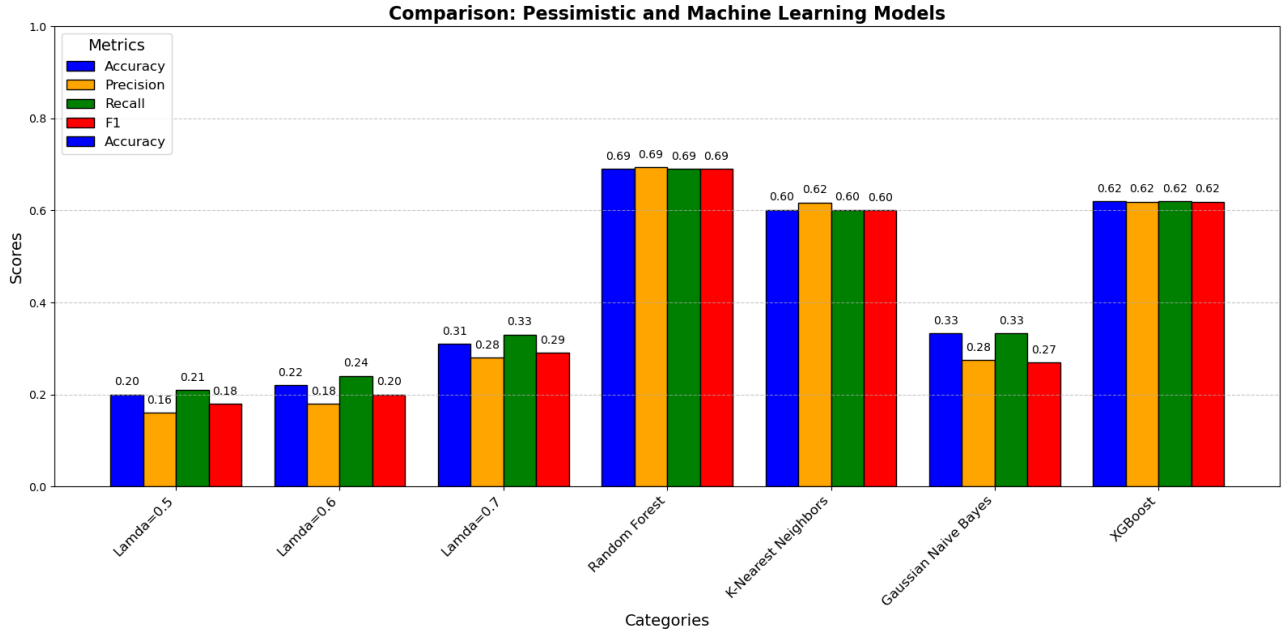

Figure 5.6: Comparasion_Electre_ML1

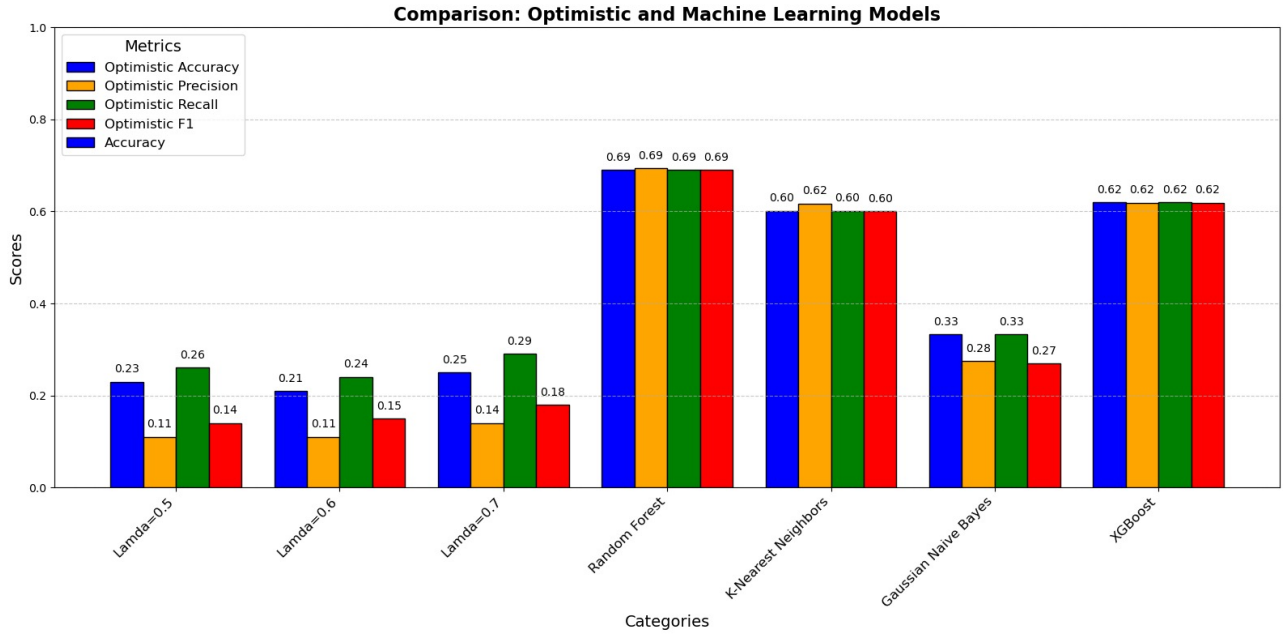

Figure 5.7: Comparasion_Electre_ML2

- **Optimistic vs. Machine Learning Models:**

- Machine learning models significantly outperformed the **Optimistic Classification** across all metrics.

- -**Random Forest** achieved the highest Accuracy (0.69) and F1-Score (0.69), compared to Optimistic's best Accuracy (0.25) and F1-Score (0.18).

- Optimistic Classification prioritized Recall (0.29 at best), but its low Precision (0.14 at best) and overall poor Accuracy made it less reliable.

- **Pessimistic vs. Machine Learning Models:**

-**Pessimistic Classification** performed better than Optimistic Classification but still fell short of machine learning models.

- At its best ($\lambda = 0.7$), Pessimistic achieved Accuracy (0.31) and F1-Score (0.29), which were significantly lower than machine learning models like Random Forest or XGBoost.

- Pessimistic Classification's conservative nature improved Precision (0.28) but sacrificed Recall compared to machine learning models.

- **Conclusion:** Machine learning models, especially Random Forest, offered superior Accuracy and F1-Score, making them the preferred choice over rule-based methods (Optimistic and Pessimistic). However, Pessimistic Classification was more reliable than Optimistic for imbalanced datasets.

5.5.2 Conclusion

The ELECTRE-Tri models provide an interpretable and structured decision-making framework, especially in contexts requiring deterministic categorizations. However, their performance lags behind that of machine learning models, such as Random Forest and XGBoost, in terms of predictive accuracy and generalization. Future work could focus on integrating ELECTRE-Tri principles with machine learning techniques to leverage the strengths of both approaches.

6 Comparison

6.1 Results

After applying the Simple Decision Rules model shown in chapter4 to the Beard Guys database[1][2] , the following results as shown in 6.1 .

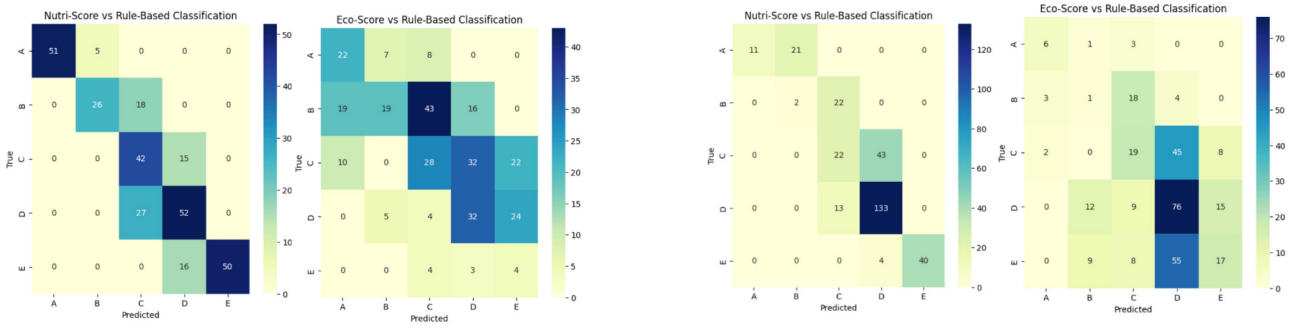

Figure 6.1: Our MCDA Model Simple Decision Rules compared Beard Guys

6.2 Final Model Estimation and Performance

In this section, we compare the Simple Decision Rules (SDR) model applied to our dataset with the results from the Beard Guys dataset using three key performance metrics:

- Consistency Rate
- Boundary Adjustment Rate
- Adjustment Distribution Consistency (B)

[1] Group of BDMA students Muhammad Qasim Khan,Gabriel Lozano Pinzon,Benjamin Gold

[2] `https://docs.google.com/spreadsheets/d/16jbL3unDd-vght0ZPOl2ctZTwaEuLSHX/edit?usp=sharing&ouid=107089449053844369351&rtpof=true&sd=true`

These metrics help assess the alignment and accuracy of the classification between the two datasets and determine how well the SDR model performs relative to Beard Guys' methodology.
The table6.1 showed results.

Comparison	Consistency Rate	Boundary Adjustment Rate	Adjustment Distribution Consistency (B)
Our Dataset	73.2%	43.9%	59.1%
Beard Guys	66.9%	40.9%	8.33%

Table 6.1: Comparison of two Datasets on the same model

6.2.1 Consistency Rate

The Consistency Rate[3] measures how consistently food products are classified across different iterations or models. In our case, the higher 73.2% consistency rate for our dataset suggests that the SDR model has a more reliable classification, indicating that the decision rules are more stable and less subject to variations. On the other hand, the Beard Guys dataset shows a lower 66.9% consistency rate, suggesting that their classification process might be more sensitive to changes in the model or data, possibly due to the complexity of their MCDA approach, which includes weighted criteria and more complex models. This result may also indicate that their model is more flexible, adjusting classifications more based on various parameters, whereas our simpler SDR model leads to more stable classifications.

6.2.2 Boundary Adjustment Rate

The Boundary Adjustment Rate refers to the frequency with which the boundaries (cutoff thresholds for classification) are adjusted between categories (e.g., from Neutral to Healthy). A higher rate indicates a more dynamic adjustment based on certain data characteristics or model flexibility.

6.2.3 Adjustment Distribution Consistency (B)

Our dataset shows a slightly higher boundary adjustment rate of 43.9%, which could indicate that the SDR model allows for more flexibility in adjusting the boundaries when classifying food items, particularly for edge cases. This may be due to the simpler, rule-based structure of the SDR model, where boundaries are adjusted more frequently in response to minor changes in the nutritional data. The Beard Guys dataset, with a rate of 40.9%, demonstrates a slightly less frequent boundary

[3] https://en.wikipedia.org/wiki/Consistency_(statistics)

adjustment, which could be due to the more robust, weighted criteria in their model, leading to fewer changes to the classification boundaries.

The Adjustment Distribution Consistency (B) is a metric that measures the consistency of how the adjustments are distributed across food items. A higher value indicates that adjustments to food classifications are more evenly spread across the dataset. This metric helps us understand the fairness and evenness of the classification process.

Our dataset shows a 59.1% consistency in adjustment distribution, which is quite high. This indicates that the SDR model distributes the adjustments in a relatively balanced way, ensuring that no particular category or group of food items is disproportionately affected by classification changes.

In contrast, the Beard Guys dataset shows a much lower 8.33% adjustment distribution consistency. This suggests that their model might apply adjustments in a more skewed manner, perhaps focusing on a specific subset of food items or being more sensitive to certain nutritional factors. This could reflect the complexity of their multi-criteria model, where different food types may be influenced more heavily by specific factors.

6.2.4 Conclusion

In conclusion, the Simple Decision Rules model applied to our dataset demonstrates a higher consistency rate and a more balanced adjustment distribution when compared to the Beard Guys dataset. However, the Beard Guys model shows a more complex decision-making process, with more nuanced boundary adjustments, though at the cost of distribution consistency.

While the SDR model provides a simpler, more consistent, and stable classification method, the Beard Guys' model offers more flexibility and sophistication, allowing for finer distinctions but with a slightly lower level of consistency. Depending on the application context, either model could be preferable—SDR for faster and more reliable classifications or Beard Guys' model for more detailed and flexible categorization.

7 Conclusion about models

7.1 Most Comfortable Model

As figure 5.6 shown,the Random Forest model was the most comfortable due to its high performance:

Accuracy (0.689873), Precision (0.693873), Recall (0.689873), and F1-Score (0.690735).

It handled imbalanced data well and achieved the best balance across metrics.

7.2 Most Suitable Model

As figure 3.26 shown,the Pessimistic ELECTRE-Tri model at $\lambda = 0.7$ was suitable for calculating Nutri-Score:

Simplicity aligned with Nutri-Score's predefined thresholds.

It achieved reasonable performance (Accuracy: 0.31, F1: 0.29) while being interpretable and easy to implement.

7.3 Most Explainable Model

As figure 3.26 shown,The Pessimistic ELECTRE-Tri model at $\lambda = 0.7$:

Transparency: Threshold-based logic was easily interpretable for consumers.

Trust: It mimicked Nutri-Score's methodology, making the computation clear and relatable.

Reference

Articles

[CMM12] Cailloux, Olivier, Patrick Meyer, and Vincent Mousseau (2012). “Eliciting electre tri category limits for a group of decision makers”. In: *European Journal of Operational Research* 223.1, pp. 133–140.

[DFR10] Dias, José Almeida, João Rui Figueira, and Bernard Roy (2010). “Electre tri-c: A multiple criteria sorting method based on characteristic reference actions”. In: *European Journal of Operational Research* 204.3, pp. 565–580.

[Dou+09] Doumpos, Michael et al. (2009). “An evolutionary approach to construction of outranking models for multicriteria classification: The case of the ELECTRE TRI method”. In: *European Journal of Operational Research* 199.2, pp. 496–505.

[Roy02] Roy, Bernard (2002). “Présentation et interprétation de la méthode ELECTRE TRI pour affecter des zones dans des catégories de risque”. In: *Document du LAMSADE* 124, p. 25.

Books

[BS02] Belton, Valerie and Theodor J. Stewart (2002). *Multiple Criteria Decision Analysis: An Integrated Approach*. Springer Science & Business Media.

[Bis06] Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Miscellaneous

[May25] Mayag, Brice (2024-2025). *Lecture Slides: Decision Modelling*. Class Lecture, Centrale-Supélec. Accessible only via Microsoft Teams.