

Decision Modeling Project

ELABORATION OF DECISION MODELS FOR NUTRI-SCORE

Joel Anil JOSE
Adrian PATRICIO

¹ Access the github link here: <https://github.com/alpatricio10/decision-modeling-project>

Contents

1 Data Collection	1
1.1 Introduction	1
1.2 Data Extraction	1
1.2.1 Data Collection Pipeline	1
1.2.2 Dataset Balancing and Validation	2
1.3 Data Quality	2
1.3.1 Exclusion of Beverages and Other Categories	2
2 Standard Nutri-Score Model	3
2.1 Standard Nutri-Score Model Implementation	3
2.1.1 Special Rule for Proteins	3
2.2 Interactive Dashboard	4
2.2.1 Features and Design	4
3 ELECTRE-TRI Model	5
3.1 Profile Selection	5
3.1.1 General Selection Details	5
3.1.2 Uniform Quintiles	5
3.1.3 Dataset Quantiles	5
3.1.4 Class Centroids	7
3.1.5 Comparison of Profile Strategies	7
3.2 Weight Selection	7
3.2.1 Equal Weights	8
3.2.2 Correlation-Based Weights	8
3.2.3 Entropy-Based Weights	8
3.2.4 Differential Evolution (Optimization)	9
3.2.5 Comparison of Weight Strategies	9
3.3 Results and Analysis	9
3.3.1 Summary of Similarity Scores	9
3.3.2 Pessimistic vs. Optimistic Procedures	10
3.3.3 Impact of Limiting Profile Selection	10
3.3.4 Impact of Weight Strategies	11
3.3.5 Threshold Sensitivity (λ)	11
3.3.6 Difference between Nutri-score Model	11
3.4 Confusion Matrix Analysis	11
3.4.1 Optimized Weights & Centroid Profiles	11
3.4.2 Equal Weights & Uniform Profiles	12
3.4.3 Distribution-Based Profiles	13

4 Weighted Sum Model	15
4.1 Methodology	15
4.2 Normalization as Utility Function	15
4.2.1 Utility of Nutrients to Limit (Negative Criteria)	15
4.2.2 Utility of Nutrients to Encourage (Positive Criteria)	16
4.3 Classification via Threshold Selection	16
4.4 Weight Selection	16
4.4.1 Equal Weights	16
4.4.2 Correlation-Based Weights	17
4.4.3 Principal Component Analysis (PCA)	17
4.4.4 Random Forest Feature Importance	17
4.4.5 Differential Evolution (Optimization)	17
4.4.6 Comparison of Weight Strategies	17
4.5 Threshold Selection	18
4.5.1 Equal Interval Thresholds	19
4.5.2 Quantile-Based Thresholds	19
4.5.3 K-Means Clustering	19
4.5.4 Jenks Natural Breaks	19
4.5.5 Gaussian Mixture Models (GMM)	19
4.5.6 Differential Evolution (Optimization)	19
4.5.7 Comparison of Threshold Strategies	20
4.6 Decision-Maker Strategies	21
4.6.1 Balanced, Sustainable Weights	21
4.6.2 Negative-Heavy Weights	22
4.7 Results and Analysis	22
4.7.1 Summary of Similarity Scores	22
4.7.2 Impact of Optimization	22
4.7.3 Impact of Threshold vs Weights	23
4.7.4 Limitations	23
4.8 Confusion Matrix Analysis	23
4.8.1 Optimized Weights & Optimized Thresholds	23
4.8.2 Baseline: Equal Weights & Equal Intervals	24
4.8.3 Random Forest & Quantile Thresholds	24
4.8.4 Decision-Maker Strategies	25
4.9 Comparison with ELECTRE-TRI	25
5 Machine Learning Models	28
5.1 Methodology	28
5.1.1 Limitation of the ML Approach	28
5.1.2 Data Preprocessing and Training	28
5.1.3 Algorithm Selection	28
5.2 Results and Analysis	29
5.2.1 Evaluation Strategy: K-Fold Cross-Validation	29
5.3 Models and Confusion Matrix Analysis	30
5.3.1 Ensemble Models (Gradient Boosting & Random Forest)	30

5.3.2	Baseline: Decision Tree	30
5.3.3	Linear and Distance-Based Models	31
5.4	Explainability	32
5.5	Future Work	32
5.6	Comparison with MCDA Models and Conclusions	33
6	Comparison with Other Group	37
6.1	Results and Analysis	37
7	Conclusion	41
7.1	Performance Summary	41
7.2	The Model We Prefer	41
7.3	Final Thoughts	41
A	Complete Confusion Matrices	42
A.1	Electre-TRI Model	42
A.2	Weighted Sum Model	53
Bibliography		58

List of Figures

1.1	Distribution of Nutri-Score and Green-Score classes in the final dataset.	2
2.1	Screenshot of the Nutri-Score Dashboard showing a computed result.	4
3.1	Distribution Graphs for each of the criteria	6
3.2	Pessimistic ($\lambda = 0.6$)	12
3.3	Pessimistic ($\lambda = 0.7$)	12
3.4	Optimistic ($\lambda = 0.6$)	12
3.5	Optimistic ($\lambda = 0.7$)	12
3.6	Confusion matrices for Optimized + Centroid models .	12
3.7	Pessimistic ($\lambda = 0.6$)	13
3.8	Pessimistic ($\lambda = 0.7$)	13
3.9	Optimistic ($\lambda = 0.6$)	13
3.10	Optimistic ($\lambda = 0.7$)	13
3.11	Confusion matrices for Equal + Uniform models .	13
3.12	Pessimistic ($\lambda = 0.6$)	14
3.13	Pessimistic ($\lambda = 0.7$)	14
3.14	Optimistic ($\lambda = 0.6$)	14
3.15	Optimistic ($\lambda = 0.7$)	14
3.16	Confusion matrices for Equal + Distribution models .	14
4.1	Confusion Matrix: Optimized Weights and Optimized Thresholds.	24
4.2	Confusion Matrix: Equal Weights and Equal Interval Thresholds.	25
4.3	Confusion Matrix: Random Forest Weights and Quantile Thresholds.	26
4.4	Balanced + Green Score Focus	27
4.5	Negative-Heavy	27
5.1	Comparison of predictions diagram.	29
5.2	Gradient Boosting.	30
5.3	Random Forest Performance.	30
5.4	Decision Tree Performance.	31
5.5	Logistic Regression Performance.	32
5.6	KNN Performance.	32
5.7	SVM (RBF) Performance.	33
5.8	Feature Importance of Tree-based Models	35
5.9	Permutation Importance of KNN Model	36
5.10	Permutation Importance of SVM Model	36
6.1	Distribution of Other Group's Dataset	37
6.2	Confusion Matrix from other group on their data	38
6.3	Confusion Matrix from our group on external data	39

6.4 Confusion Matrix from our group using the optimized weighted sum model on external data	40
A.1 Pessimistic ($\lambda = 0.6$)	42
A.2 Pessimistic ($\lambda = 0.7$)	42
A.3 Optimistic ($\lambda = 0.6$)	42
A.4 Optimistic ($\lambda = 0.7$)	42
A.5 Confusion matrices for Correlation + Centroid models.	42
A.6 Pessimistic ($\lambda = 0.6$)	43
A.7 Pessimistic ($\lambda = 0.7$)	43
A.8 Optimistic ($\lambda = 0.6$)	43
A.9 Optimistic ($\lambda = 0.7$)	43
A.10 Confusion matrices for Correlation + Distribution models.	43
A.11 Pessimistic ($\lambda = 0.6$)	44
A.12 Pessimistic ($\lambda = 0.7$)	44
A.13 Optimistic ($\lambda = 0.6$)	44
A.14 Optimistic ($\lambda = 0.7$)	44
A.15 Confusion matrices for Correlation + Equal models.	44
A.16 Pessimistic ($\lambda = 0.6$)	45
A.17 Pessimistic ($\lambda = 0.7$)	45
A.18 Optimistic ($\lambda = 0.6$)	45
A.19 Optimistic ($\lambda = 0.7$)	45
A.20 Confusion matrices for Entropy + Centroid models.	45
A.21 Pessimistic ($\lambda = 0.6$)	46
A.22 Pessimistic ($\lambda = 0.7$)	46
A.23 Optimistic ($\lambda = 0.6$)	46
A.24 Optimistic ($\lambda = 0.7$)	46
A.25 Confusion matrices for Entropy + Distribution models.	46
A.26 Pessimistic ($\lambda = 0.6$)	47
A.27 Pessimistic ($\lambda = 0.7$)	47
A.28 Optimistic ($\lambda = 0.6$)	47
A.29 Optimistic ($\lambda = 0.7$)	47
A.30 Confusion matrices for Entropy + Equal models.	47
A.31 Pessimistic ($\lambda = 0.6$)	48
A.32 Pessimistic ($\lambda = 0.7$)	48
A.33 Optimistic ($\lambda = 0.6$)	48
A.34 Optimistic ($\lambda = 0.7$)	48
A.35 Confusion matrices for Optimized + Centroid models.	48
A.36 Pessimistic ($\lambda = 0.6$)	49
A.37 Pessimistic ($\lambda = 0.7$)	49
A.38 Optimistic ($\lambda = 0.6$)	49
A.39 Optimistic ($\lambda = 0.7$)	49
A.40 Confusion matrices for Optimized + Distribution models.	49
A.41 Pessimistic ($\lambda = 0.6$)	50
A.42 Pessimistic ($\lambda = 0.7$)	50

A.43 Optimistic ($\lambda = 0.6$)	50
A.44 Optimistic ($\lambda = 0.7$)	50
A.45 Confusion matrices for Equal + Centroid models	50
A.46 Pessimistic ($\lambda = 0.6$)	51
A.47 Pessimistic ($\lambda = 0.7$)	51
A.48 Optimistic ($\lambda = 0.6$)	51
A.49 Optimistic ($\lambda = 0.7$)	51
A.50 Confusion matrices for Equal + Distribution models	51
A.51 Pessimistic ($\lambda = 0.6$)	52
A.52 Pessimistic ($\lambda = 0.7$)	52
A.53 Optimistic ($\lambda = 0.6$)	52
A.54 Optimistic ($\lambda = 0.7$)	52
A.55 Confusion matrices for Equal + Equal models	52
A.56 Correlation Weights + Equal.	53
A.57 Random Forest Weights + Equal.	53
A.58 PCA Weights + Equal.	53
A.59 Equal Weights + Equal.	53
A.60 Correlation Weights + Quantile.	54
A.61 Random Forest Weights + Quantile.	54
A.62 PCA Weights + Quantile.	54
A.63 Equal Weights + Quantile.	54
A.64 Correlation Weights + Jenks.	54
A.65 Random Forest Weights + Jenks.	54
A.66 PCA Weights + Jenks.	55
A.67 Equal Weights + Jenks.	55
A.68 Correlation Weights + Kmeans.	55
A.69 Random Forest Weights + Kmeans.	55
A.70 PCA Weights + Kmeans.	55
A.71 Equal Weights + Kmeans.	55
A.72 Correlation Weights + GMM.	56
A.73 Random Forest Weights + GMM.	56
A.74 PCA Weights + GMM.	56
A.75 Equal Weights + GMM.	56
A.76 Balanced, Sustainable Weights + Quantile.	56
A.77 Negative-Heavy Weights + Quantile.	56
A.78 Optimized	57

CHAPTER 1

Data Collection

1.1 Introduction

Nutri-Score is a nutritional labeling system adopted by several European countries to help consumers make healthier food choices. It classifies food products into five categories, ranging from **A** for the highest nutritional quality to **E** for the lowest.

Our project aims to model this decision process using different Multi-Criteria Decision Analysis (MCDA) approaches. The goal is not to replicate the official Nutri-Score algorithm perfectly, nor to propose a better alternative. Instead, we aim to explore how different MCDA models diverge from or converge with the official model and analyze the sensitivity of the nutritional classification to different assumptions. To achieve this, a balanced dataset is required as a prerequisite.

1.2 Data Extraction

The data extraction process collects nutritional information from the **Open Food Facts** database [2], a collaborative open-data project containing millions of food products. We designed a custom extraction pipeline to create a dataset that meets the constraints of the project.

1.2.1 Data Collection Pipeline

Data is gathered via the Open Food Facts API using a category-based search strategy. This ensures we cover a wide variety of food groups rather than relying on random sampling. The pipeline targets specific solid food categories such as *breads, cereals, prepared meats, cheese, and plant-based meals*, while explicitly excluding beverages to maintain consistency with the solid-food scoring algorithm.

The extraction logic iterates through the selected categories and fetches products in batches. To respect API rate limits, a delay of 0.5 seconds is enforced between requests. For each product, we extract the following attributes based on the project requirements:

- **Identification:** Product name and brand.
- **Nutri-Score Criteria (per 100g):** Energy, Saturated Fat, Sugars, Salt, Proteins, Fiber, Fruits, Vegetable and Legumes Percatage.
- **Labels:** Nutri-Score grade (A-E) and Green-Score grade (A-E).

The raw JSON response is parsed, ensuring that we only retain products with complete nutritional profiles.

1.2.2 Dataset Balancing and Validation

A raw dump from the API often results in an imbalanced dataset (for instance, an over-representation of some categories). To satisfy the project requirement of at least 15% representation for each Nutri-Score class and 10% for each Green-Score class, we implemented a balancing algorithm to automatically downsample over-represented classes. This resulted in the distribution as shown in Figure 1.1, which falls within the required constraints.

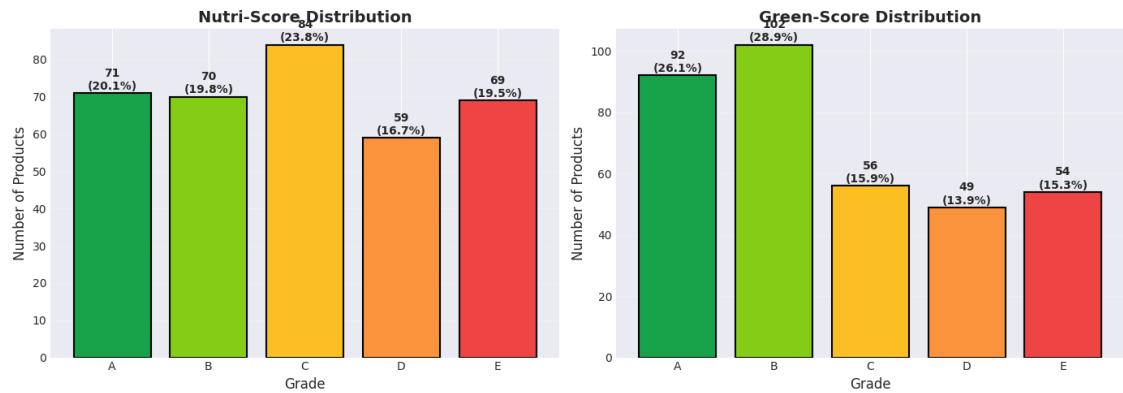


Figure 1.1: Distribution of Nutri-Score and Green-Score classes in the final dataset.

Some other validations we implemented was to check for completeness to ensure there are no missing values for the criteria and cardinality checks to verify that the total count exceeds 200 products. **All in all, the dataset contains 353 products**, which is above the minimum number of products. The final output is exported to an Excel file as required.

1.3 Data Quality

During the verification phase, we observed discrepancies for a subset of products where our computed score did not match the official label from Open Food Facts. We had to consider these when collecting our dataset.

1.3.1 Exclusion of Beverages and Other Categories

To minimize these calculation variances, we restricted our dataset to general solid foods. **We explicitly excluded beverages**, since the algorithm for beverages differs significantly from solid foods (e.g., non-nutritious sweeteners criteria is added). Despite this, some noise remains in the labels possibly due to the version mismatch between the project's formula and the database. For the purpose of this project, we treat the computed values based on the project statement's computation as our reference for building the MCDA models, and only products with matching values are kept.

CHAPTER 2

Standard Nutri-Score Model

2.1 Standard Nutri-Score Model Implementation

The standard Nutri-Score algorithm computes a nutritional score derived from seven quantitative criteria per 100g of product. This calculation is divided into two main components: a negative component N (nutrients to limit) and a positive component P (nutrients to encourage).

The scoring logic assigns points as follows, consistent with the specific thresholds provided in the course project guidelines [1]:

- **Negative Component (N):** Sum of points for Energy (kJ , 0–10), Saturated Fatty Acids (g , 0–10), Sugars (g , 0–15), and Salt (g , 0–20). The maximum theoretically possible value for N is 55.
- **Positive Component (P):** Sum of points for Proteins (g , 0–5), Fiber (g , 0–5), and Fruits/Vegetables/Legumes (FVL , %, 0–5). The maximum possible value for P is 15.

The final nutritional score S is calculated as:

$$S = \sum N - \sum P \quad (2.1)$$

2.1.1 Special Rule for Proteins

A conditional rule applies when the total negative score is high. If the negative component $N \geq 11$ and the fruit/vegetable content (FVL) is less than 80%, the protein points are excluded from the calculation [1]. In this case, the score becomes:

$$S = \sum N - (\text{Fiber points} + \text{FVL points})$$

This rule prevents products high in sugar or saturated fats from receiving a favorable score solely due to protein content. The final score S is mapped to a class (A–E) using the standard thresholds defined in the project specification:

Score Range	Class	Color
[−15, 0]	A	Dark Green
[1, 2]	B	Light Green
[3, 10]	C	Yellow
[11, 18]	D	Light Orange
[19, 40]	E	Red

Our Python implementation utilizes a series of threshold-based `if-elif` chains to strictly adhere to these rules. We validated this implementation against our database of 353 solid food products and tested it against the labels provided by Open Food Facts [2].

2.2 Interactive Dashboard

To fulfill project requirement 4.2 [1], we developed a graphical user interface (GUI) that enables real-time Nutri-Score computation. This tool serves as both a verification utility and an educational demonstrator of the algorithm's piecewise-linear logic.

2.2.1 Features and Design

The dashboard enables users to input custom nutritional values for the seven required criteria. Upon submission, the system instantly calculates:

- The total **Negative (N)** and **Positive (P)** sub-scores.
- The final numeric **Nutri-Score**.
- The corresponding **Nutri-Score Label (A–E)** with official color coding.

The application is hosted online and available at:

<https://v0-nutri-score-calculator-app.vercel.app/>. A sample of the dashboard input and output is shown in Figure 2.1.

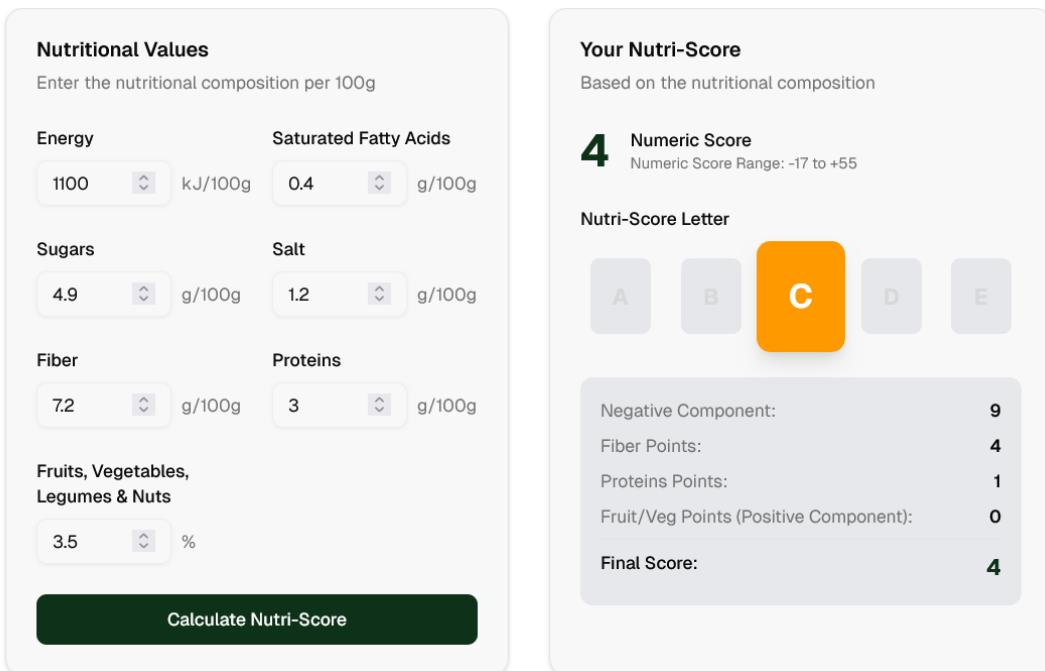


Figure 2.1: Screenshot of the Nutri-Score Dashboard showing a computed result.

CHAPTER 3

ELECTRE-TRI Model

3.1 Profile Selection

In the MR-Sort model, the limiting profiles π^1, \dots, π^6 define the upper and lower boundaries, as well as the boundaries between the ordered classes (A to E). We experimented with three distinct statistical approaches to determine these boundaries.

3.1.1 General Selection Details

For all profiles, the minimum and maximum observed values are taken as either the lower and upper bounds depending on the criterion. For criteria that need to be maximized, the profile values are increasing, showing that higher values yield better overall scores. On the other hand, criteria that need to be minimized have profile values that are decreasing, showing that lower values yield better overall scores. For the green score criterion, the lower and upper bounds are fixed to the theoretical minimum of 0 and the theoretical maximum of 100.

3.1.2 Uniform Quintiles

This method assumes that the value range of each criterion should be divided into five equal intervals. It operates on the assumption that nutritional quality is uniformly distributed across the range of the nutrients. The profiles are shown in Table 3.1

- **Methodology:** We compute the distribution of each criterion in the dataset and partition it into five equal-sized groups, each containing 20% of the products. These distributions are shown visually in 3.1. The values at the boundaries of these groups are used to define the corresponding profiles.
- **Implication:** This ignores the actual skew of nutritional data (e.g., most products have low salt, but the range extends to 100g).

3.1.3 Dataset Quantiles

This approach derives profiles based on the actual distribution of data within our collected database. It assumes that the distribution of Nutri-Score classes in the world follows the distribution observed in our sample. The profiles are shown in Table 3.2

- **Methodology:** We compute the Nutri-score distribution as shown in Figure 1.1. The values at the boundaries of each score are used to define the corresponding profiles. For instance, score E has 19.5% of the data, so the value that lies at this percentile is used for the profile.



Figure 3.1: Distribution Graphs for each of the criteria

	energy	saturated fat	sugars	salt	proteins	fiber	fruit	green score
π^1	86	0	0	0	0	0	0	0
π^2	428.6	0.1	0.9	0.02	1.2	0	0	36
π^3	791.6	0.5	2.58	0.47	5.38	0.6	0	54
π^4	1252	1.12	4.2	0.74	8	2	2.5	70
π^5	1643.2	3	25.44	1.19	12	4.96	34.77	78
π^6	3033	34	98.5	17.2	35	17.1	100	100

Table 3.1: Profiles for uniform quintiles

- **Implication:** This is dependent on how much data is available during the modeling phase, so this might not extend properly if the model is applied to larger, more varied datasets.

	energy	saturated fat	sugars	salt	proteins	fiber	fruit	green score
π^1	86	0	0	0	0	0	0	0
π^2	430.2	0.1	0.9	0.02	12	4.92	34.7	78
π^3	788.2	0.5	2.57	0.47	8	2.0	2.5	70
π^4	1392.4	1.5	4.94	0.8	4.66	0.5	0	50
π^5	1648.2	3	26.8	1.2	1.2	0	0	35
π^6	3033	34	98.5	17.2	35	17.1	100	100

Table 3.2: Profiles for dataset quintiles

3.1.4 Class Centroids

This method assumes that the boundary between two classes lies at the midpoint between the "average" product for each of the classes. The profiles are shown in Table 3.3

- **Methodology:** We calculate the centroid (mean vector) for all products labeled 'A', 'B', etc., in the ground truth dataset. The profile π^k is set as the arithmetic mean of the centroids of class C^k and C^{k+1} .
- **Implication:** This attempts to mimic the existing clustering of the official Nutri-Score but adapts it to the specific density of our dataset. Similar to the dataset quintiles, this is hugely dependent on the current distribution.

3.1.5 Comparison of Profile Strategies

Table 3.4 shows a summary of the different profile selection strategies.

3.2 Weight Selection

The relative importance of the criteria significantly alters the classification behavior. We investigated four weight computation strategies to see how it affects the model.

	energy	saturated fat	sugars	salt	proteins	fiber	fruit	green
π^1	86	0	0	0	0	0	0	0
π^2	1546.9	5.47	29.17	1.37	6.18	2.00	12.92	54.06
π^3	1106.4	2.35	15.36	0.84	6.19	2.52	17.19	59.12
π^4	800.6	1.22	4.75	0.72	7.38	2.52	17.19	59.12
π^5	800.6	0.74	3.03	0.52	9.29	2.85	20.61	60.51
π^6	3033	34	98.5	17.2	35	17.1	100	100

Table 3.3: Profiles for centroids

Method	Pros	Cons
Uniform Quintiles	- Simple and fully unsupervised. - Provides a stable baseline. - Easy to interpret and reproduce.	- Sensitive to extreme values. - Ignores real-world product distribution.
Distribution-based Quintiles	- Reflects market structure. - Adapts to skewed data.	- Dataset-dependent. - Sensitive to sampling bias. - Profiles may shift with new data.
Class Centroids	- Incorporates class-specific patterns. - Less affected by single outliers.	- Requires reliable labeled data. - Circular dependency on ground truth. - Sensitive to class overlap.

Table 3.4: Pros and cons of profile selection methods

3.2.1 Equal Weights

This baseline approach assigns the same importance to every criterion, regardless of its nutritional impact or statistical variance. Each of the eight criteria is given a weight of exactly 12.5%.

3.2.2 Correlation-Based Weights

Weights are derived from the Pearson correlation coefficient between each criterion and the official Nutri-Score. This assumes that criteria which historically correlate strongly with the final score should have higher influence. The correlations are weighted relative to each other and normalized to [0, 1] to get the final weights.

3.2.3 Entropy-Based Weights

This method assigns weights based on the information content (divergence) of the data. Criteria with high variability across the dataset are given higher weights, as they provide more discriminatory power between products. The entropies are weighted relative to each other and normalized to [0, 1] to get the final weights.

3.2.4 Differential Evolution (Optimization)

We utilized an optimization algorithm given by a the differential evolution library to learn the weights that maximize the similarity between the MR-Sort classification and the official Nutri-Score. This attempts to replicate the Nutri-score using MCDA logic.

3.2.5 Comparison of Weight Strategies

Table 3.5 shows a summary of the different weight selection strategies, while Table 3.6 shows the computed weights for each strategy.

Strategy	Pros	Cons
Equal Weights	- No bias; treats all nutrients as equally critical.	- Ignores scientific reality (e.g. sugar is worse than salt in nutri-score logic).
Correlation	- Captures linear relationships with the target.	- Fails to capture non-linear threshold effects.
Entropy	- Data-driven, and highlights distinguishing features.	- May overemphasize noisy criteria with high variance but low health impact.
Optimization	- Maximizes agreement with the reference scores.	- Risk of overfitting to the training dataset. - Does not really give explainable weight selections.

Table 3.5: Pros and Cons of Weight Elicitation Methods.

Method	energy	sat. fat	sugars	salt	proteins	fiber	fruit	green score
Equal Weights	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
Correlation-Based	0.196	0.130	0.189	0.098	0.103	0.120	0.065	0.098
Entropy-Based	0.043	0.203	0.164	0.146	0.065	0.134	0.217	0.025
Differential Evolution	0.141	0.125	0.141	0.125	0.046	0.178	0.197	0.046

Table 3.6: Criterion weights under different weight selection strategies

3.3 Results and Analysis

In this section, we analyze how the combinations of these profiles and weights affect the model's behavior. We emphasize that **a lower similarity score does not necessarily imply a worse model, but rather one that values nutritional attributes differently than the standard Nutri-Score**. This extends to the other models as well.

3.3.1 Summary of Similarity Scores

To evaluate the similarity of our MR-Sort models with the official Nutri-Score, we tested every combination of parameters. This includes three profile selection methods, four weighting strategies, and two majority thresholds ($\lambda = 0.6$ and $\lambda = 0.7$) across both

pessimistic and optimistic sorting procedures. Tables 3.7 and 3.8 summarize the similarity scores. The highest similarity is highlighted in **bold**.

Weight Strategy	Uniform Profiles		Distribution Profiles		Centroid Profiles	
	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.6$	$\lambda = 0.7$
Equal Weights	0.43	0.43	0.31	0.31	0.49	0.49
Correlation	0.37	0.38	0.41	0.36	0.46	0.31
Entropy	0.42	0.30	0.33	0.35	0.35	0.28
Optimized	0.39	0.46	0.34	0.40	0.50	0.47

Table 3.7: Similarity Scores: **Pessimistic Sorting**

Weight Strategy	Uniform Profiles		Distribution Profiles		Centroid Profiles	
	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.6$	$\lambda = 0.7$
Equal Weights	0.31	0.31	0.31	0.31	0.45	0.45
Correlation	0.31	0.30	0.34	0.26	0.40	0.30
Entropy	0.26	0.30	0.32	0.28	0.44	0.33
Optimized	0.27	0.28	0.32	0.31	0.35	0.33

Table 3.8: Similarity Scores: **Optimistic Sorting**

The comparative results in Tables 3.7 and 3.8 reveal significant divergence between the MR-Sort models and the Nutri-Score. By analyzing the similarity scores, we can derive several key insights regarding the sorting logic.

3.3.2 Pessimistic vs. Optimistic Procedures

The most prominent trend is the higher similarity of the **pessimistic** sorting procedure over the optimistic approach. Across nearly all weight and profile configurations, the pessimistic assignment yields higher similarity scores (peaking at **0.50**) compared to the optimistic assignment (peaking at 0.45).

It seems that the standard Nutri-Score system seems to be more safe and conservative with regards to nutrition, where a product is not classified as 'A' unless it performs well in most of the criteria. The pessimistic models kind of replicates this by requiring the majority consensus to move to a better class. In contrast, the optimistic procedure frequently assigns products to higher classes, which leads to lower similarities.

3.3.3 Impact of Limiting Profile Selection

- **Centroid Profiles:** The Centroid-based profiles consistently outperformed the Uniform and Distribution methods. This suggests that the official Nutri-score is not necessarily uniform distributed. Instead, the products form distinct clusters, and using the centroids allows the model to anchor its boundaries to the actual centers of these clusters.
- **Distribution Profiles:** The quantile-based method performed poorly. This implies that the distribution of nutritional quality is skewed. By forcing the model to assign

20% of products to each classm it contradicts the ground truth (where 'D' and 'E' products might be over-represented in certain categories like cookies).

3.3.4 Impact of Weight Strategies

- **Equal Weights:** Surprisingly, the equal weights strategy performed good, achieving 0.49 accuracy in the Pessimistic/Centroid configuration, which is nearly identical to the optimized result (0.50). This suggests that the seven Nutri-Score criteria are relatively balanced in their contribution to the final classification.
- **Optimization:** While the optimized strategy achieved the highest similarity, it did not generalize well to the Optimistic procedure. This indicates the optimization over-fitted to the specific logic of the pessimistic rule.

3.3.5 Threshold Sensitivity (λ)

Comparing $\lambda = 0.6$ and $\lambda = 0.7$, the model generally performs better at the lower threshold ($\lambda = 0.6$). Increasing λ to 0.7 typically reduced accuracy.

A threshold of 0.7 makes the veto effect too strong. The official Nutri-Score is fully compensatory (i.e. a very bad value in Sugar can be offset by very good Fiber). MR-Sort is non-compensatory, where setting λ too high essentially forbids this compensation, causing the model to downgrade products too aggressively compared to the official score.

3.3.6 Difference between Nutri-score Model

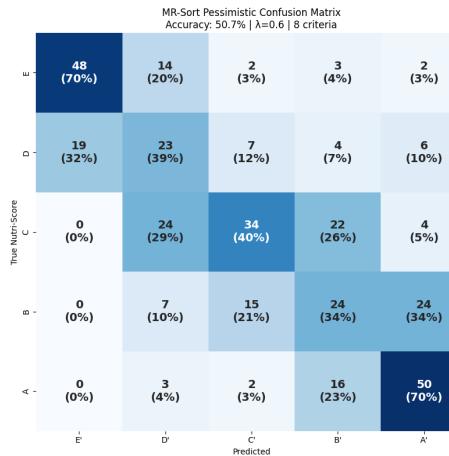
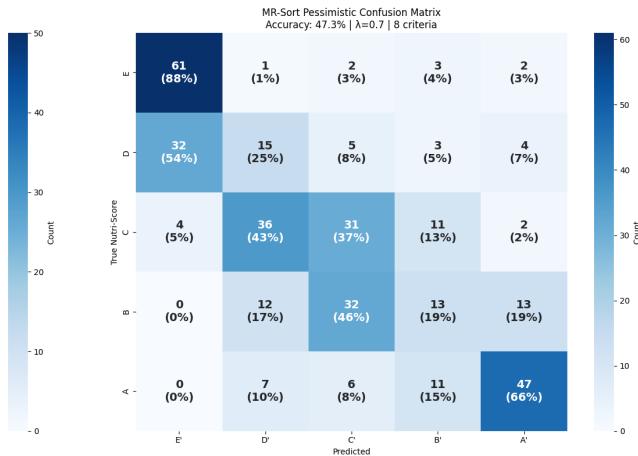
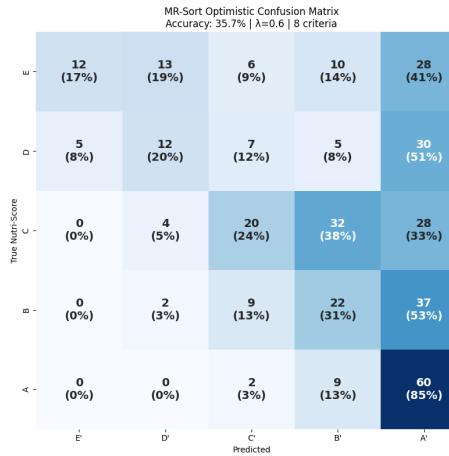
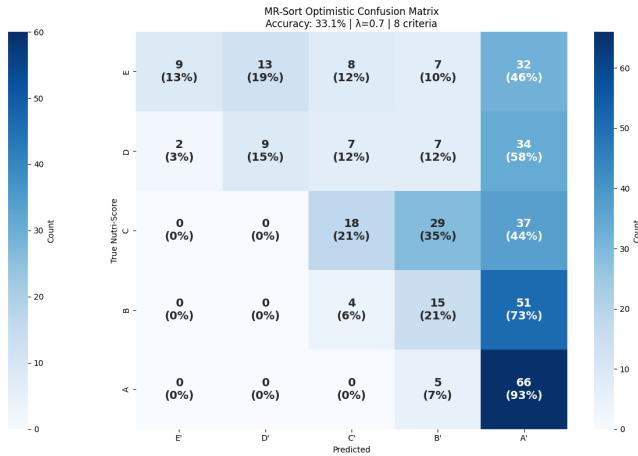
The fact that our best model plateaus at 50% similarity highlights a theoretical difference. The Nutri-Score is an arithmetic, compensatory model (i.e. the points are summed), while MR-Sort is a majority-rule, non-compensatory model which cannot easily replace the Nutri-score perfectly. In fact, the similarity shows that the MR-Sort model is essentially a random guess in comparison to the official Nutri-score at the best case, and even worse than random in the other cases. This highlights the fundamental difference in the priorities of this model compared to Nutri-score.

3.4 Confusion Matrix Analysis

The final step in our evaluation is a granular analysis of the classification behavior using confusion matrices. These matrices visualize the similarity between our model and the official Nutri-Score. In these grids, the rows represent the official Nutri-Score, and the columns represent our model classifications. We have selected configurations that highlight the impact of profile selection and the non-compensatory nature of the MR-Sort logic. For a full set of matrices, see Appendix A.

3.4.1 Optimized Weights & Centroid Profiles

The combination of Differential Evolution for weight computation and centroids for profile boundaries yielded our highest similarity score (**50.0%**).

Figure 3.2: Pessimistic ($\lambda = 0.6$)Figure 3.3: Pessimistic ($\lambda = 0.7$)Figure 3.4: Optimistic ($\lambda = 0.6$)Figure 3.5: Optimistic ($\lambda = 0.7$)Figure 3.6: Confusion matrices for **Optimized + Centroid models**.

These matrices demonstrate the model's ability to anchor categories to real-world data clusters. The model shows strong diagonal performance for Classes A and E, confirming that the very healthy and very unhealthy profiles are distinct enough for our model. Increasing λ from 0.6 to 0.7 causes a visible downward shift in classifications. A higher threshold requirement makes the model more conservative, frequently downgrading products to lower scores than their official counterparts. As obvious from above, the optimistic setup is heavily skewed to classify products as A, while the pessimistic setup is more balanced.

3.4.2 Equal Weights & Uniform Profiles

This naive setup assumes all nutrients are equally important and that categories are evenly spaced across the value range. It achieved a surprisingly high similarity.

While the similarity is high, looking more into the matrix reveals significant skewing

3.4. Confusion Matrix Analysis

13

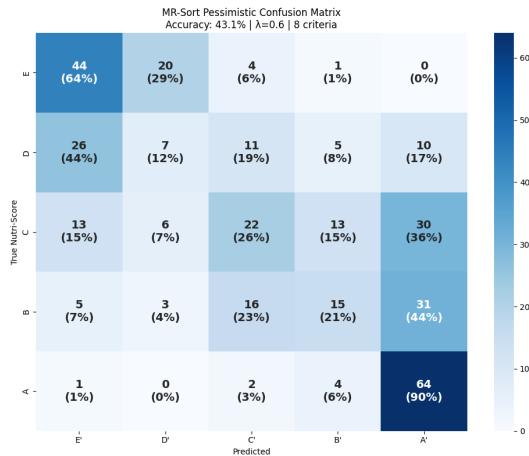


Figure 3.7: Pessimistic ($\lambda = 0.6$)

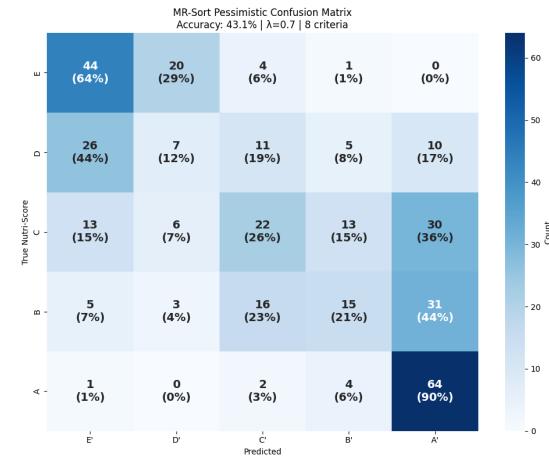


Figure 3.8: Pessimistic ($\lambda = 0.7$)

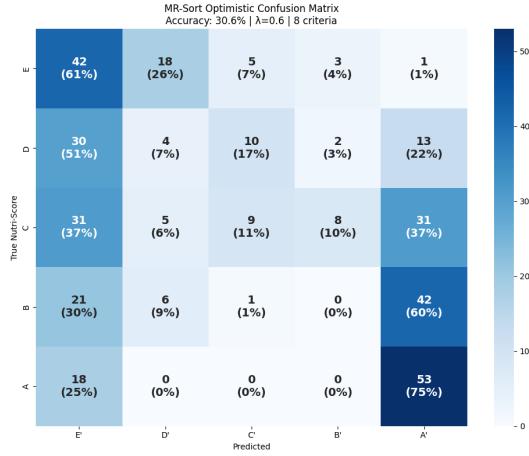


Figure 3.9: Optimistic ($\lambda = 0.6$)

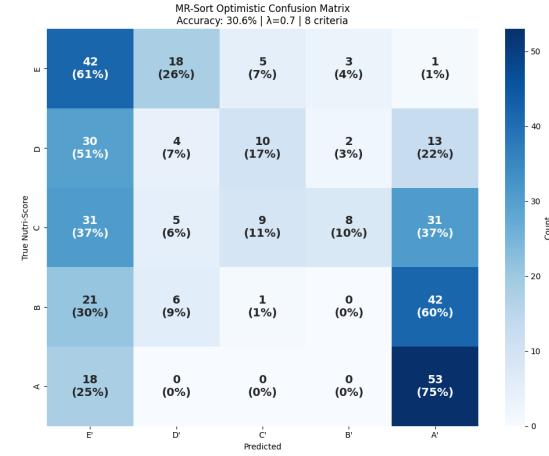


Figure 3.10: Optimistic ($\lambda = 0.7$)

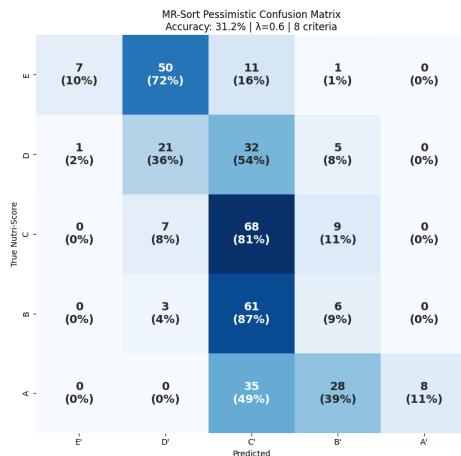
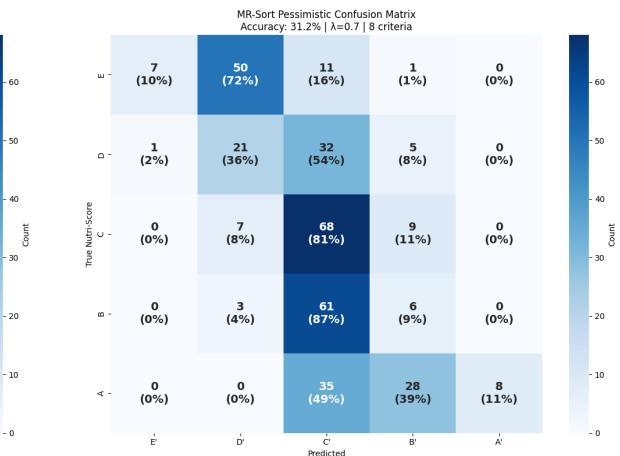
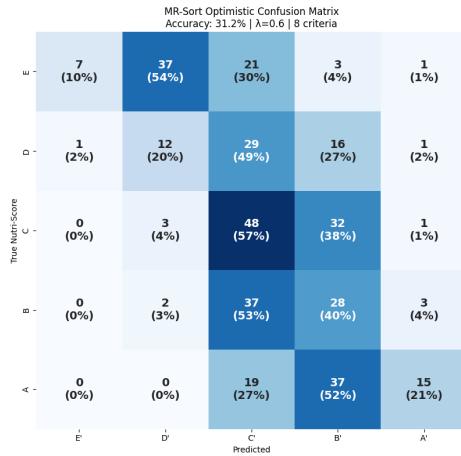
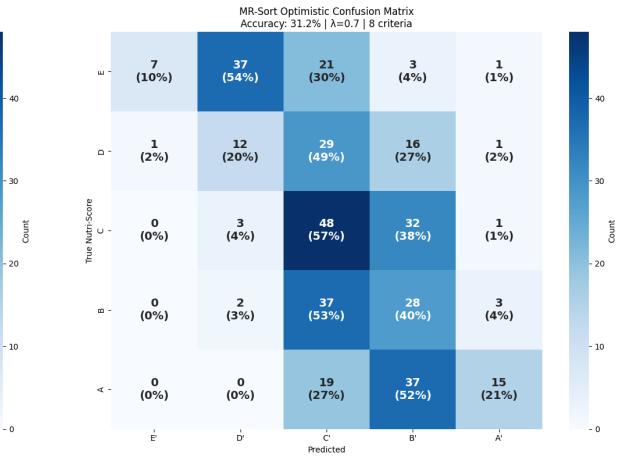
Figure 3.11: Confusion matrices for Equal + Uniform models.

in the classification, where most of the products are classified as either A or E for all cases. It only got a high similarity because it got most of those edge classes correctly. This setup fails to capture any middle-ground nuances for the Nutri-score, which suggests that nutrition is not evenly-spaced. This also suggests that this model is not a good model for Nutri-score since it does not give any significant information about nutrition if most of the products end up in the edge classes anyway.

3.4.3 Distribution-Based Profiles

By forcing the profiles to follow the dataset's quantiles (assigning 20% of products to each class), the model surprisingly achieved its lowest similarity (0.31).

As opposed to the previous case where all the classifications are in the edges, this model is skewed towards the middle classes instead. For the pessimistic model, most of the classifications are in class C', while the optimistic model is more balanced between B'

Figure 3.12: Pessimistic ($\lambda = 0.6$)Figure 3.13: Pessimistic ($\lambda = 0.7$)Figure 3.14: Optimistic ($\lambda = 0.6$)Figure 3.15: Optimistic ($\lambda = 0.7$)Figure 3.16: Confusion matrices for **Equal + Distribution models.**

and C'. Since this got a low score, it suggests that Nutri-score is primarily an absolute model based on just the score, and not on how the products rank relative to each other which is what the distribution method implies.

CHAPTER 4

Weighted Sum Model

4.1 Methodology

The Weighted Sum Model evaluates food products by aggregating multiple nutritional and environmental criteria into a single holistic index. The model follows a linear aggregation approach:

$$F = \sum_{i=1}^n w_i u_i(x_i)$$

where:

- w_i is the normalized weight of criterion i
- x_i is the raw performance of a product on criterion i .
- $u_i(\cdot)$ is the **marginal utility function** for criterion i .

One main limitation is that **we only considered positive weights for our criteria**. For future work, it would be interesting to explore negative weights as well, since this would more accurately represent the subtractive effect of the negative criteria to our Nutri-score model. This would also be more similar to how the official Nutri-score model is devised.

4.2 Normalization as Utility Function

In our model, the utility function u_i is given by a **Min-Max Normalization procedure**. This ensures comparability across criteria with different units (e.g., grams vs. kilojoules),

4.2.1 Utility of Nutrients to Limit (Negative Criteria)

For energy, saturated fat, sugars, and salt, higher values yield lower utility. The utility function represents a minimization logic:

$$u_i(x) = \frac{\max(x) - x}{\max(x) - \min(x)}$$

Here, a product with the theoretical maximum amount receives a utility of 0, while the minimum amount receives a utility of 1.

4.2.2 Utility of Nutrients to Encourage (Positive Criteria)

For proteins, fiber, fvl percentage, and the green score, higher values yield higher utility. The utility function follows a maximization logic:

$$u_j(x) = \frac{x - \min(x)}{\max(x) - \min(x)}$$

In this case, a product with the theoretical minimum amount receives a utility of 0, while the maximum amount receives a utility of 1.

4.3 Classification via Threshold Selection

Once a weighted sum is calculated, the final step is to map this continuous numerical value (ranging from 0 to 1) into one of the five ordered Nutri-Score classes (A', B', C', D', E'). This is achieved by defining four numerical thresholds (t_1, t_2, t_3, t_4) that act as the boundaries between classes. The classification logic follows a descending order of utility:

- **Class A':** $F \geq t_4$ (Highest Utility)
- **Class B':** $t_4 > F \geq t_3$
- **Class C':** $t_3 > F \geq t_2$
- **Class D':** $t_2 > F \geq t_1$
- **Class E':** $F < t_1$ (Lowest Utility)

By adjusting these thresholds, we can control how strictly the model assigns products to the different categories. This allows our model to maintain a consistent decision logic even when applied to different food databases.

4.4 Weight Selection

Five distinct methods were implemented to determine the weights of the criteria. These methods allow us to derive weights based on different data-driven perspectives rather than relying on arbitrary assignments.

4.4.1 Equal Weights

This baseline approach assigns the same importance to every criterion, regardless of its nutritional impact or statistical variance. Each of the eight criteria is given a weight of exactly 12.5%.

This is useful because it provides a completely neutral starting point without any bias. It serves as a benchmark to evaluate whether more complex weighting strategies actually improve the model's alignment with nutritional goals.

4.4.2 Correlation-Based Weights

Weights are derived from the absolute Pearson correlation coefficients between each criterion and the original Nutri-Score labels. Criteria that exhibit a stronger linear relationship with the official score are granted higher importance.

This method is useful because it directly measures how much each individual nutrient influences the final official grade. It ensures that nutrients which are known to be primary drivers of health labels in existing systems are given a proportional weight in our model.

4.4.3 Principal Component Analysis (PCA)

This unsupervised approach assigns weights based on the amount of information or variance each nutrient contributes to the dataset. By analyzing the first principal component, weights were derived from the absolute influence of each feature on that component.

It is useful here because it identifies which nutrients vary the most across different food products. It allows the model to prioritize criteria that are most effective at distinguishing one product from another based on the actual spread of the data.

4.4.4 Random Forest Feature Importance

A Random Forest classifier was trained on the ground truth data. Weights were assigned according to the Gini importance. This highlights which nutrients are most decisive in helping the algorithm correctly separate products into their final classes.

This method is applicable because it accounts for complex, non-linear relationships between nutrients. Unlike simple correlations, it can identify if a nutrient is important only when combined with others, providing a more complex understanding of food composition.

4.4.5 Differential Evolution (Optimization)

Optimization was used to find the weights and score thresholds that minimize the difference between our model and the official Nutri-Score. This method mathematically searches for the specific set of parameters that results in the highest level of agreement.

This is applicable because it serves as the ultimate benchmark for alignment. It allows us to closely match the existing Nutri-Score system by fine-tuning the balance of every criterion simultaneously.

4.4.6 Comparison of Weight Strategies

Table 4.1 shows a summary of the different weight selection strategies.

Strategy	Pros	Cons
Equal Weights	- Fully unbiased and simple. - Provides a stable baseline for comparison.	- Ignores nutrition information. - Not adapted to the dataset structure.
Correlation	- Identifies direct links between a nutrient and the final grade. - Very easy to interpret and explain to non-experts.	- Misses nutrients that work together in groups. - Assumes the relationship is a simple straight line.
PCA	- Reflects the natural variety found in food products. - Removes overlapping information between similar nutrients.	- High variety in a nutrient does not always mean it is healthy. - The first component might ignore subtle but vital health factors.
Random Forest	- Recognizes that some nutrients are only dangerous in specific amounts. - Handles noisy or messy data very well.	- Can favor nutrients that appear in many different forms in the data. - The logic behind the final weights can be difficult to trace.
Optimization	- Finds the exact mathematical solution for the best possible match. - Adjusts weights and thresholds at the same time.	- May produce weights that do not make sense from a nutrition perspective. - It is highly specific to the products used during the search.

Table 4.1: Pros and Cons of Weight Elicitation Methods for Weighted Sum Model.

The weights generated by each method are summarized in Table 4.2. Notably, both optimization and random forest methods put the greatest emphasis on salt. PCA focused heavily on energy (21.1%), while correlation focused on sugars (21.6%). This shows that each method emphasizes different criteria in categorizing the food products.

Method	Energy	Sat. Fat	Sugars	Salt	Prot.	Fiber	FVL	Green
Optimization	0.068	0.213	0.138	0.409	0.076	0.071	0.017	0.007
Random Forest	0.149	0.137	0.164	0.170	0.113	0.097	0.089	0.079
PCA	0.211	0.193	0.024	0.015	0.143	0.074	0.190	0.149
Correlation	0.191	0.163	0.216	0.107	0.090	0.075	0.075	0.082
Equal	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125

Table 4.2: Criterion weights across different elicitation strategies.

4.5 Threshold Selection

After calculating the weighted sum of each product, we must establish thresholds to classify the products into different categories. We implemented six methods to determine these thresholds.

4.5.1 Equal Interval Thresholds

This method divides the entire range of scores into five equal-sized intervals. This is applicable as a baseline approach. It assumes that each grade represents exactly 20% of the possible utility spectrum. It is useful for understanding how products are distributed when the categories are not influenced by the dataset.

4.5.2 Quantile-Based Thresholds

This method assumes that the distribution of products across classes should be relatively balanced based on the observed utility scores in the dataset.

It is useful when the priority is to maintain a competitive ranking within a specific market. By calculating the 20th, 40th, 60th, and 80th percentiles, the model ensures that each class is populated, preventing a skewed classification where most products end up in a single category.

4.5.3 K-Means Clustering

K-Means is a centroid-based clustering algorithm that partitions the scores into five distinct groups by minimizing the variance within each cluster.

This is applicable because it identifies natural breaks in the nutritional quality of products. Instead of forcing equal group sizes, it groups products that possess similar utility scores, effectively finding the natural clusters of healthy vs unhealthy items in the data.

4.5.4 Jenks Natural Breaks

The Jenks method is designed to determine the best arrangement of values into different classes by reducing the variance within classes and maximizing the variance between classes.

This is highly relevant for Nutri-Score modeling because it is specifically designed to handle data that is not uniformly distributed. It identifies boundaries where there are significant gaps in the nutritional utility of the food products.

4.5.5 Gaussian Mixture Models (GMM)

GMM is a probabilistic model that assumes the utility scores are generated from a mixture of several Gaussian distributions with unknown parameters.

Unlike K-Means, which uses hard boundaries, GMM allows for a more flexible, probabilistic approach to thresholding. It is useful here because it can account for overlapping nutritional profiles, assigning thresholds based on the most likely distribution a product belongs to.

4.5.6 Differential Evolution (Optimization)

Similar to the weight selection process, this method treats the thresholds as parameters to be optimized to maximize the agreement with the official labels. In this case, the

thresholds are **0.731**, **0.789**, **0.814** and **0.830** when using the optimized weights as well.

4.5.7 Comparison of Threshold Strategies

Table 4.3 shows a summary of the different threshold selection strategies.

Strategy	Pros	Cons
Equal Interval	<ul style="list-style-type: none"> - Extremely simple and easy to communicate. - Independent of dataset skew or bias. 	<ul style="list-style-type: none"> - Often results in empty classes if data is clustered. - Does not account for natural breaks in nutrition.
Quantile	<ul style="list-style-type: none"> - Ensures all classes are represented. - Not affected by extreme outliers. 	<ul style="list-style-type: none"> - Assumes the market is perfectly balanced, which is rarely true in nutrition.
K-Means	<ul style="list-style-type: none"> - Groups products by their actual similarity in utility score. 	<ul style="list-style-type: none"> - Can be sensitive to the initial starting points (centroids).
Jenks	<ul style="list-style-type: none"> - Best at finding real gaps in the data. - Highly accurate for skewed nutritional data. 	<ul style="list-style-type: none"> - Computationally intensive for very large datasets.
GMM	<ul style="list-style-type: none"> - Accounts for the density and shape of the score distribution. - More flexible than standard K-Means. 	<ul style="list-style-type: none"> - Requires the data to roughly follow a bell-curve shape within clusters.
Optimization	<ul style="list-style-type: none"> - Mathematically guarantees the highest alignment with Nutri-Score. 	<ul style="list-style-type: none"> - May create thresholds that are too specific to one dataset to be reused.

Table 4.3: Pros and Cons of Threshold Selection Mechanisms.

The following tables summarize the thresholds generated by each method. Since the thresholds are dependent on the weighted sum values, they vary depending on what weight strategy is used.

Method	t_1 (E/D)	t_2 (D/C)	t_3 (C/B)	t_4 (B/A)
Quantile	0.495	0.554	0.590	0.630
Jenks	0.424	0.516	0.590	0.664
K-Means	0.427	0.521	0.591	0.665
GMM	0.432	0.522	0.590	0.663

Table 4.4: Calculated threshold boundaries for each selection strategy with equal weights

Method	t_1 (E/D)	t_2 (D/C)	t_3 (C/B)	t_4 (B/A)
Quantile	0.558	0.636	0.661	0.691
Jenks	0.471	0.568	0.645	0.702
K-Means	0.475	0.573	0.645	0.706
GMM	0.479	0.579	0.649	0.707

Table 4.5: Calculated threshold boundaries for each selection strategy with RF weights

Method	t_1 (E/D)	t_2 (D/C)	t_3 (C/B)	t_4 (B/A)
Quantile	0.442	0.501	0.534	0.586
Jenks	0.316	0.450	0.531	0.627
K-Means	0.325	0.452	0.532	0.627
GMM	0.346	0.481	0.589	0.711

Table 4.6: Calculated threshold boundaries for each selection strategy with PCA weights

Method	t_1 (E/D)	t_2 (D/C)	t_3 (C/B)	t_4 (B/A)
Quantile	0.580	0.668	0.698	0.724
Jenks	0.459	0.569	0.657	0.735
K-Means	0.467	0.572	0.659	0.736
GMM	0.479	0.590	0.671	0.740

Table 4.7: Calculated threshold boundaries for each selection strategy with Correlation-based weights

4.6 Decision-Maker Strategies

In addition to the statistical and optimized methods described above, we acted as decision-makers to manually define weight profiles. This allowed us to test out specific nutritional weights and observe how the model behavior shifts when prioritizing certain criteria over others. For simplicity, we only tested this out with the quantile thresholds.

4.6.1 Balanced, Sustainable Weights

This strategy adopts a holistic view of the product, with balanced weights for the traditional Nutri-score criteria, but putting a greater emphasis on the sustainability as given by the green score.

Parameter	Energy	Sat. Fat	Sugars	Salt	Prot.	Fiber	FVL	Green
Weight	0.10	0.10	0.10	0.10	0.10	0.10	0.15	0.25

Table 4.8: Weight distribution for the balanced, sustainable strategy

Thresholds	t_1 (E/D)	t_2 (D/C)	t_3 (C/B)	t_4 (B/A)
Value	0.454	0.517	0.581	0.635

Table 4.9: Quantile-based thresholds for balanced classification

This approach treats the environmental footprint as the single most important individual criterion while maintaining an even balance among all nutritional attributes. It is designed for the consumer who values overall moderation and ecological sustainability equally.

4.6.2 Negative-Heavy Weights

This strategy prioritizes risk mitigation. It focuses heavily on the nutrients to limit (i.e. Saturated Fats, Sugars, Salt, and Energy) which collectively account for 60% of the decision weight.

Parameter	Energy	Sat. Fat	Sugars	Salt	Prot.	Fiber	FVL	Green
Weight (w_j)	0.15	0.15	0.15	0.15	0.05	0.05	0.10	0.20

Table 4.10: Weight distribution for the negative-heavy strategy

Thresholds	t_1 (E/D)	t_2 (D/C)	t_3 (C/B)	t_4 (B/A)
Value	0.587	0.637	0.697	0.739

Table 4.11: Quantile-based thresholds for negative-heavy classification

This follows a risk-averse score, where the presence of harmful elements is considered more important than the presence of beneficial ones. In this logic, a high-fiber product cannot easily hide the fact that it is extremely high in salt or sugar.

4.7 Results and Analysis

In this section, we analyze how the combinations of these thresholds and weights affect the model's behavior.

4.7.1 Summary of Similarity Scores

To evaluate the consistency of the weighted sum model across various configurations, we measured the similarity between our classifications and the official Nutri-Score labels. Table 4.12 summarizes these findings.

4.7.2 Impact of Optimization

The optimized configuration achieved a similarity of 79.3%. This is the highest score across all decision models in this study, which is expected since the algorithm tries to mathematically model the Nutri-score as close as possible. It indicates that while the official Nutri-Score is complex, it is largely an additive compensatory model that can be mathematically approximated when the precise non-linear boundaries are learned by an algorithm.

Strategies	Equal	Quantile	Jenks	K-Means	GMM	Optimized
Equal Weights	0.221	0.422	0.368	0.371	0.377	-
Correlation	0.218	0.436	0.374	0.382	0.405	-
PCA	0.263	0.331	0.312	0.312	0.331	-
Random Forest	0.212	0.450	0.411	0.411	0.402	-
Balanced, Sustainable	-	0.323	-	-	-	-
Negative-Heavy	-	0.385	-	-	-	-
Optimized	-	-	-	-	-	0.793

Table 4.12: Similarity between official Nutri-score and Weighted Sum models. The columns represent each of the threshold strategies, while the rows represent the weight strategies.

4.7.3 Impact of Threshold vs Weights

An interesting finding is changing the thresholds has a far greater impact on similarity than changing the weights. Shifting from equal interval to quantile thresholds consistently doubled similarity scores, whereas shifting between correlation and random forest weights only resulted in marginal changes. This suggests that products often maintain their relative utility ranking across different weight sets, but the the class depends mostly on the strictness of the boundaries.

4.7.4 Limitations

The weighted sum approach inherently assumes a linear relationship between input and output. However, even our best data-driven baselines (Random Forest + Quantile) peaked at only 45% similarity, which is worse than guessing randomly. This suggests that the official Nutri-score is not necessarily linear, and the fact that the optimized version does not get higher results also shows that there are non-linearities that cannot be easily captured by a weighted sum.

4.8 Confusion Matrix Analysis

The final step in our evaluation is an analysis of the models using confusion matrices. These matrices visualize the similarity between our weighted sum model and the official Nutri-Score more in-depth. In these graphs, the rows represent the official Nutri-Score, and the columns represent our model predictions. We selected only the best and most interesting models for this analysis. For a full set of matrices, see Appendix A.

4.8.1 Optimized Weights & Optimized Thresholds

This configuration utilizes Differential Evolution to align both the importance of nutrients and the class boundaries with the official algorithm. It achieved the highest similarity, which is expected as mentioned in our discussion above. The confusion matrix is shown in Figure 4.1.

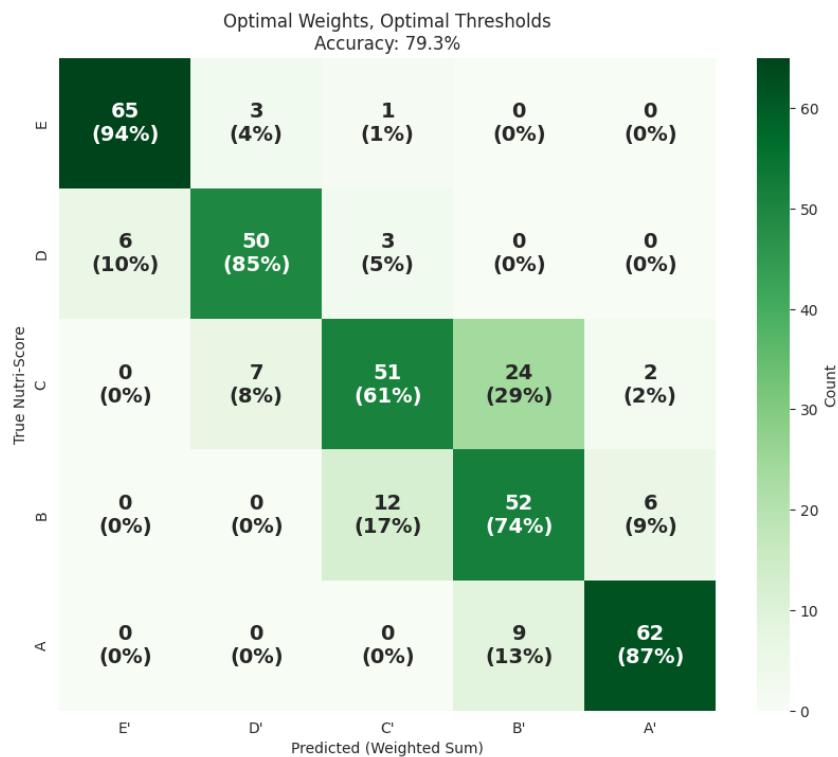


Figure 4.1: Confusion Matrix: Optimized Weights and Optimized Thresholds.

The model identifies Class E products with nearly 100% recall. This confirms that the optimized weights effectively isolate the most healthy and unhealthy products. Almost all misclassifications are in the middle classes, which suggest that most of the differences between this model and the official Nutri-score happen in between.

4.8.2 Baseline: Equal Weights & Equal Intervals

To understand the complexity of the Nutri-Score, we test a naive model where all criteria are treated equally and the intervals are evenly split. The confusion matrix is shown in Figure 4.2.

This model results in a massive accumulation of products in the middle categories, with no products reaching Class A or E. This proves that the official Nutri-Score is not exactly linear. The threshold required to reach an A or the drop to an E is much larger than a 20% slice, showing that the official system uses non-uniform thresholds.

4.8.3 Random Forest & Quantile Thresholds

This configuration uses machine learning to rank nutrient importance and statistical percentiles to set boundaries. It provides a balanced distribution without the need for global optimization, and it got the second highest similarity to the official Nutri-score classifications. The confusion matrix is shown in Figure 4.2.

Because quantiles force an even distribution, the model populates every class. Interestingly, most matches occur in class E, while the higher classes are a bit more different,

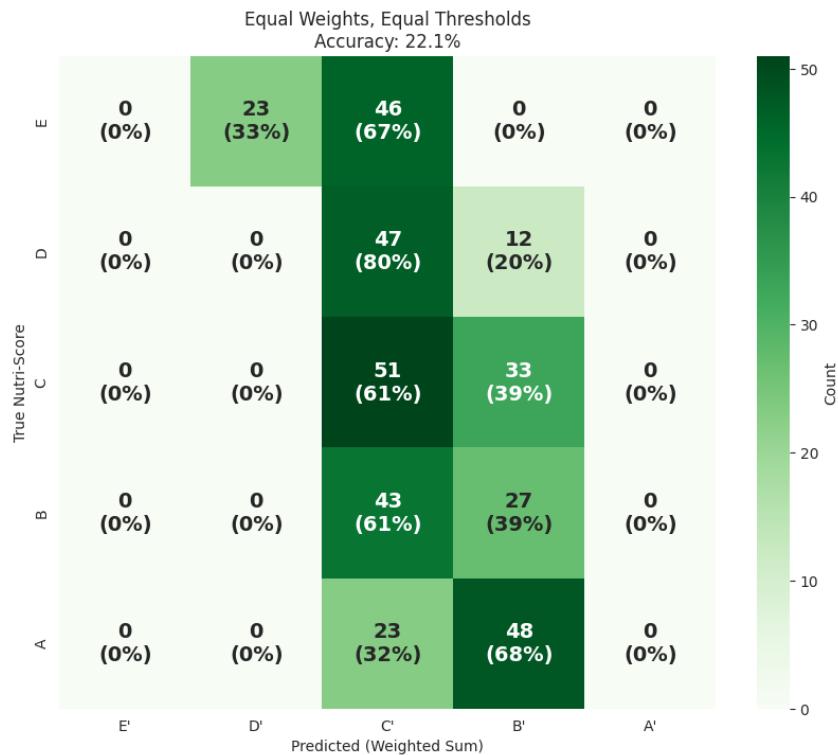


Figure 4.2: Confusion Matrix: Equal Weights and Equal Interval Thresholds.

with only at most 51% similarity for class A. This shows that the feature importances for RF don't exactly correspond to the importances in the official model.

4.8.4 Decision-Maker Strategies

These matrices represent our custom weight strategies, as shown in Figure ??.

In the Negative-Heavy matrix, we see products being re-classified into lower classes as compared to the official score. For instance, products officially labeled "B" or "C" are being reclassified by our model as "D". This reflects the stricter penalty for negative criteria compared to the official system.

For the balanced, sustainable matrix, the high weight on the green score allows products with high sustainability but average nutrition to maintain a higher class than they would in the official model. Given this, the similarity is expected to be quite low at 32.3% since there is a totally new criterion being considered heavily here. Quite noticeably, there are a lot of diverging classifications across all classes for this setup.

4.9 Comparison with ELECTRE-TRI

The weighted sum model serves as a compensatory counterpart to the non-compensatory ELECTRE-TRI model described in the previous chapter.

- **Compensation:** In the weighted sum, a very poor environmental score can be entirely offset by excellent nutritional values. In ELECTRE-TRI, a poor score in

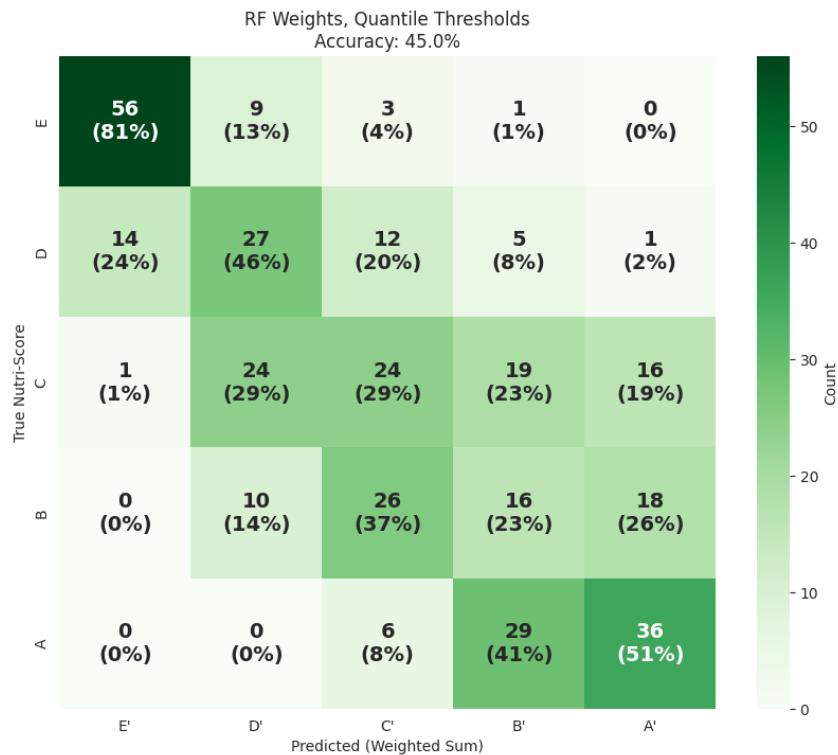


Figure 4.3: Confusion Matrix: Random Forest Weights and Quantile Thresholds.

a majority of criteria results in a lower classification regardless of performance in certain criterion.

- **Similarity:** The weighted sum reached a much higher peak similarity with the Nutri-Score than the best ELECTRE-TRI model. This highlights that the official Nutri-Score algorithm is fundamentally an additive, compensatory model.

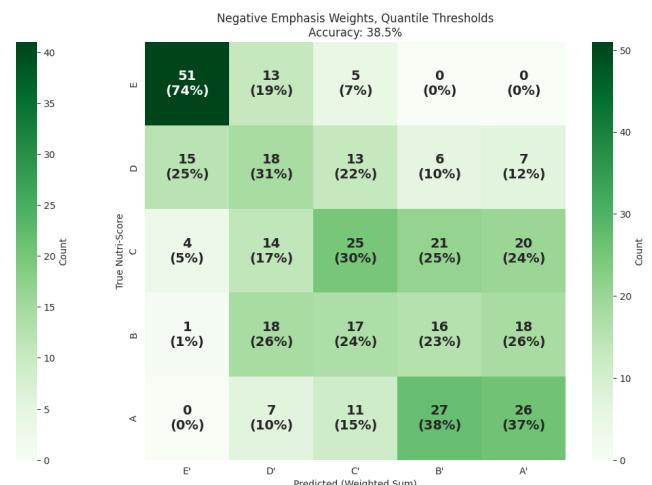
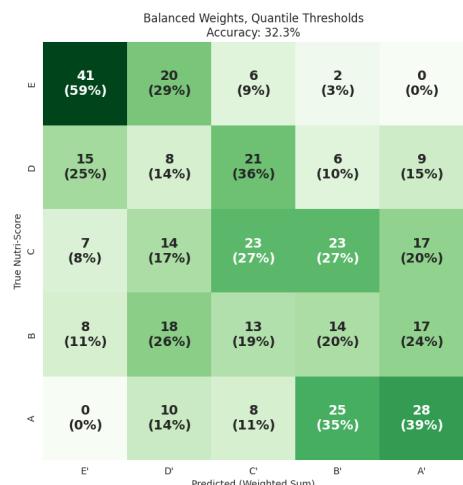


Figure 4.4: Balanced + Green Score Focus

Figure 4.5: Negative-Heavy

CHAPTER 5

Machine Learning Models

5.1 Methodology

In contrast to ELECTRE-TRI or the Weighted Sum Model, this chapter explores a purely data-driven approach. We leverage supervised learning to approximate the decision logic of the existing Nutri-Score system.

5.1.1 Limitation of the ML Approach

It is critical to note that Machine Learning models, by definition, require a **ground truth** dataset to function. In this context, the models are trained to mimic the official Nutri-Score labels provided in our database. Consequently, the primary limitation of this approach is that it cannot function as a unique, independent decision model.

While models like ELECTRE-TRI can be built on external nutritional expertise to create a new standard, ML models are bound to the biases of the training data. This approach serves as a **statistical approximation** of the current system rather than a new decision-making framework.

5.1.2 Data Preprocessing and Training

The training pipeline involved the following steps to ensure data integrity and model convergence:

- **Feature Engineering:** The model utilized 8 primary features: energy, saturated fat, sugars, salt, proteins, fiber, fvl percentage, and the green score.
- **Scaling:** Since algorithms like SVM and KNN are sensitive to the magnitude of input values, we applied **Standardization** (StandardScaler) to the features, ensuring each has a mean of 0 and a standard deviation of 1.
- **Target Encoding:** The categorical labels (A, B, C, D, E) were encoded into numerical format for the learners while maintaining their ordinal relationship.

5.1.3 Algorithm Selection

To find the most consistent approximation, we implemented and compared six different classification architectures:

- **Gradient Boosting & Random Forest:** Ensemble tree-based methods chosen for their ability to capture non-linear interactions.

- **Decision Tree:** A simple, interpretable tree structure used to visualize hierarchical decision rules.
- **Logistic Regression:** A baseline probabilistic model used to test for linear separability between classes.
- **K-Nearest Neighbors (KNN):** A distance-based classifier evaluating local similarity in the nutritional space.
- **Support Vector Machine (SVM):** A kernel-based method (RBF kernel) to find optimal hyperplanes between classes.

5.2 Results and Analysis

5.2.1 Evaluation Strategy: K-Fold Cross-Validation

To obtain a robust estimate of how well our models approximate the Nutri-Score logic, we employed **5-Fold Stratified Cross-Validation**. This approach involves splitting the dataset into five equal parts; the model is trained on four and tested on the fifth, repeating this five times so every product is used for testing exactly once. We used *stratification* to ensure that each fold maintains the same percentage of A, B, C, D, and E labels as the original dataset, preventing bias in the predictions.

Model	Mean Similarity	Standard Deviation
Gradient Boosting	74.8%	± 3.8%
Random Forest	72.8%	± 4.1%
Decision Tree	64.6%	± 4.7%
Logistic Regression	59.5%	± 4.5%
K-Nearest Neighbors	44.8%	± 4.6%
Support Vector Machine	39.1%	± 2.1%

Table 5.1: Comparison of predictions across different ML models using 5-Fold CV.

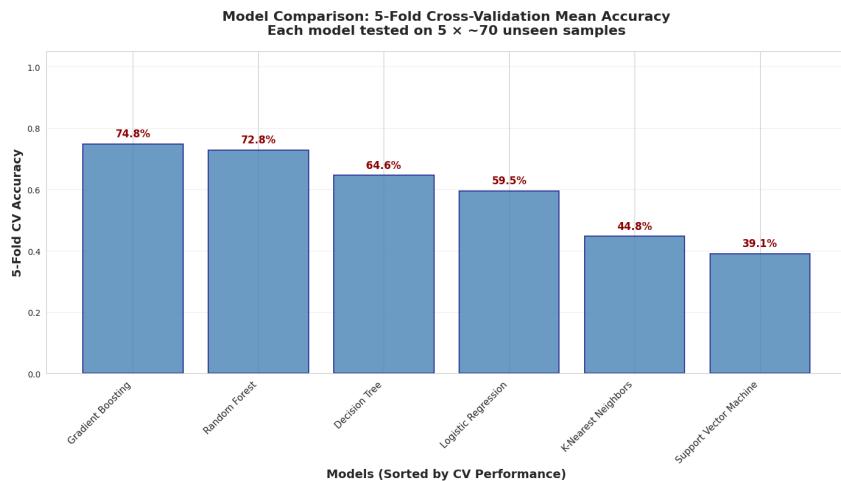


Figure 5.1: Comparison of predictions diagram.

The **Gradient Boosting** classifier achieved the highest consistency (74.8%). This indicates that the boosting process, which sequentially builds trees to minimize the residual errors of previous ones, effectively learns the specific Nutri-score boundaries that simpler models like Logistic Regression ignore.

5.3 Models and Confusion Matrix Analysis

5.3.1 Ensemble Models (Gradient Boosting & Random Forest)

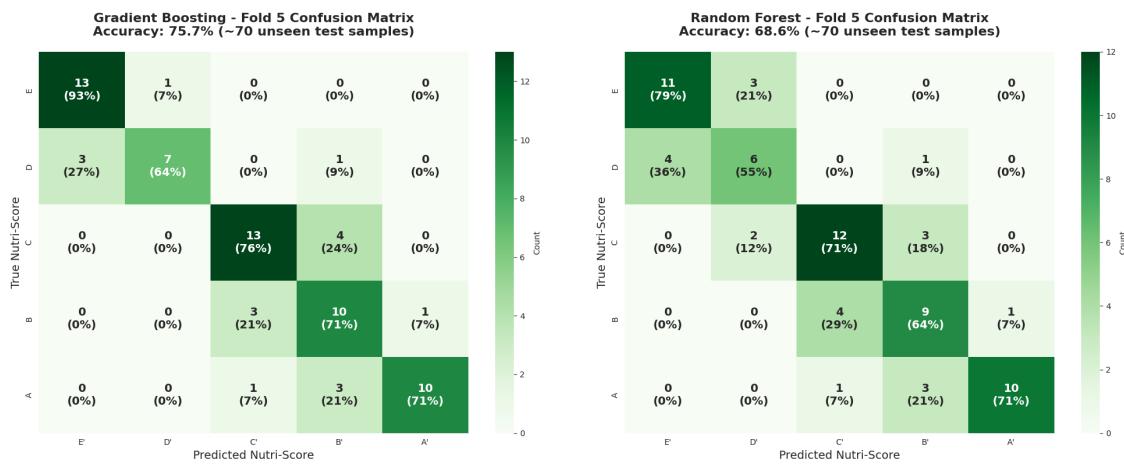


Figure 5.2: Gradient Boosting.

Figure 5.3: Random Forest Performance.

The ensemble architectures achieved the highest similarity scores, suggesting that the Nutri-Score behaves much like a high-dimensional, rule-based system. Both Gradient Boosting and Random Forest exhibit high precision when identifying products at the extremes of the spectrum, specifically for Class A and Class E. This confirms that very healthy and very unhealthy products have distinct nutritional patterns that these algorithms can isolate with high reliability. However, a significant amount of confusion remains visible in the middle-class corridor between B, C, and D. This implies that the narrow point-scoring differences used in the center of the nutritional scores are non-linear and difficult to resolve perfectly without explicit knowledge of the official point-counting formulas.

5.3.2 Baseline: Decision Tree

While the Decision Tree has a lower mean consistency (64.6%) than the ensemble methods, it provides the most transparent view of how the features are partitioned.

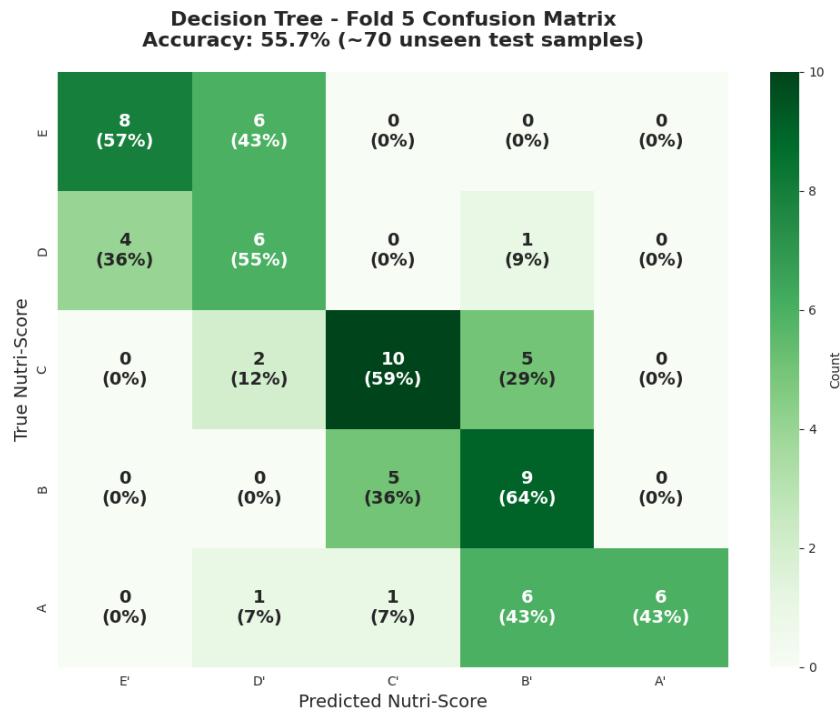


Figure 5.4: Decision Tree Performance.

The Decision Tree serves as a structural bridge between simple linear models and complex ensemble methods, providing a transparent view of how nutritional features are partitioned. While it offers more interpretability than the boosting models, it shows significantly more noise. We frequently observe instances where products are misclassified by two full grades, such as an official Class A being predicted as a Class C. This high variance suggests that a single tree structure is too prone to overfitting specific patterns in the training data and lacks the predictive power required to handle the complex, compensatory nature of the Nutri-Score algorithm.

5.3.3 Linear and Distance-Based Models

The linear and distance-based architectures, including Logistic Regression, K-Nearest Neighbors, and Support Vector Machines, performed significantly worse than the tree-based models. The failure of Logistic Regression to achieve high consistency indicates that the Nutri-Score classes are not linearly separable, meaning simple weighted sums of raw values cannot replicate the grade boundaries without non-linear transformations. Similarly, the poor performance of the Support Vector Machine and K-Nearest Neighbors proves that nutritional classes are not well-separated by simple distances or hyperplanes. The confusion matrices for these models show significant error, confirming that these models are not particularly suited to nutritional scoring.

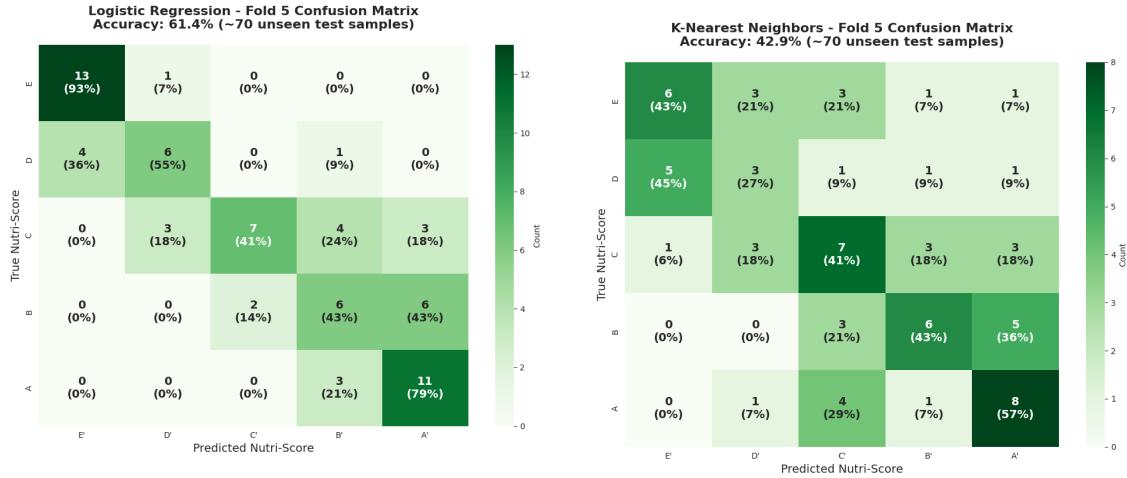


Figure 5.5: Logistic Regression Performance.

Figure 5.6: KNN Performance.

5.4 Explainability

To understand the decision logic learned by the models, we analyzed the global feature importance and marginal effects of the input variables. For the tree-based ensemble learners, we utilized the Gini importance to rank the influence of each nutrient. This analysis reveals that Salt and Sugars consistently hold the highest importance, followed by Energy and Saturated Fat. This alignment demonstrates that the machine learning models have effectively identified the core negative nutrients that drive the official Nutri-Score algorithm. The Green-Score exhibits the lowest importance most of the time, which is expected since this model tries to mimic the original Nutri-score algorithm which focuses strictly on nutrition and does not consider the Green-score in the first place. The plots are shown in Figure 5.8.

However, the analysis of distance-based and kernel-based models, such as K-Nearest Neighbors and Support Vector Machines, reveals a different structural priority through permutation importance. In these models, energy is the most influential feature, followed by FVL percentage and sugars. This indicates that within the high-dimensional nutritional space, the distance between products is primarily defined by a different set of criteria as compared to decision trees. The high importance of energy in these models confirms that they are heavily reliant on calories to distinguish between the categories. The plots are shown in Figures 5.9 and 5.10.

5.5 Future Work

While the current models provide a baseline for approximating nutritional scores, there are several directions that could be considered in the future.

- **Integration of Large Language Models (LLMs):** It would be highly valuable to investigate the potential of ChatGPT or other LLMs in nutritional assessment. Future work could test if LLMs can generate high-quality Nutri-Score labels based

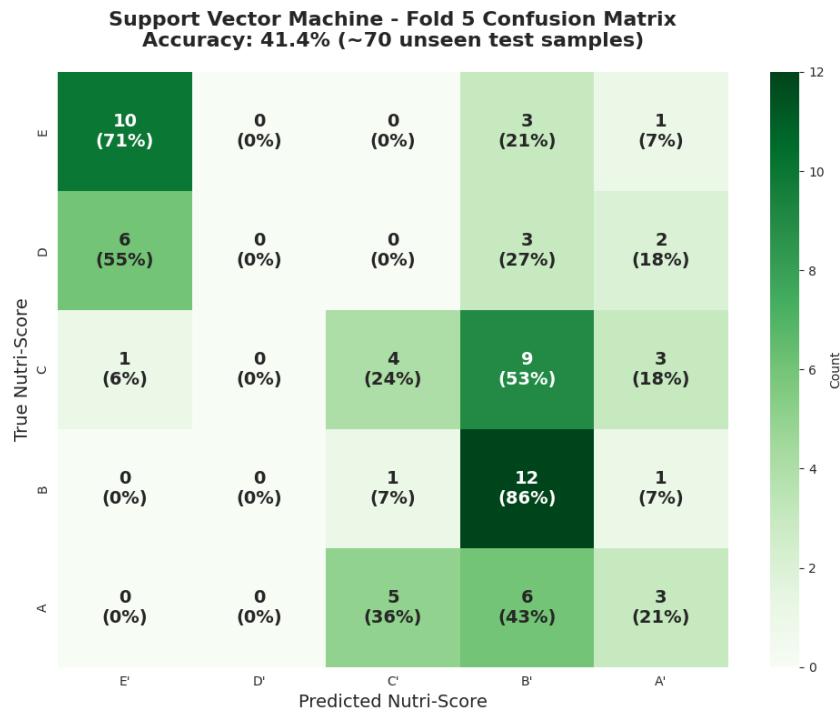


Figure 5.7: SVM (RBF) Performance.

on ingredient lists rather than just raw nutrient values. However, as noted in this study, such a move would likely come at the expense of explainability compared to the current frameworks.

- **Advanced Feature Engineering:** This project utilized raw nutrient values directly. Future iterations could involve engineering features to capture complex non-linearities to improve the consistency of the middle-class predictions (B, C, and D).
- **Alternative Architectures:** Exploring deep learning models, eXtreme Gradient Boosting (XGBoost) or other models with hyperparameter tuning could further push the similarity shown in this report.

5.6 Comparison with MCDA Models and Conclusions

The ML models effectively mimic data patterns but fail to provide a nutritionally grounded explanation for their decisions. In contrast, the ELECTRE-TRI and Weighted Sum models, though exhibiting lower similarity in certain configurations, offer a framework where boundaries are supplemented by expert knowledge. The inability of ML to implement explicit veto effects makes it less suitable for public safety labeling, where a single critical nutrient (like high amounts of salt) should theoretically trigger a downgrade regardless of other positive attributes. While ML can replicate Nutri-score through pattern recognition, the MCDA models provide a balance of consistency and explainability for real-world applications.

Interestingly, the optimized weighted sum model still got a higher similarity value than the machine learning model with the highest similarity. While this is most likely a consequence of insufficient data during training or bad data engineering, this could also suggest that Nutri-Score is ideally an arithmetic-based, compensatory system.

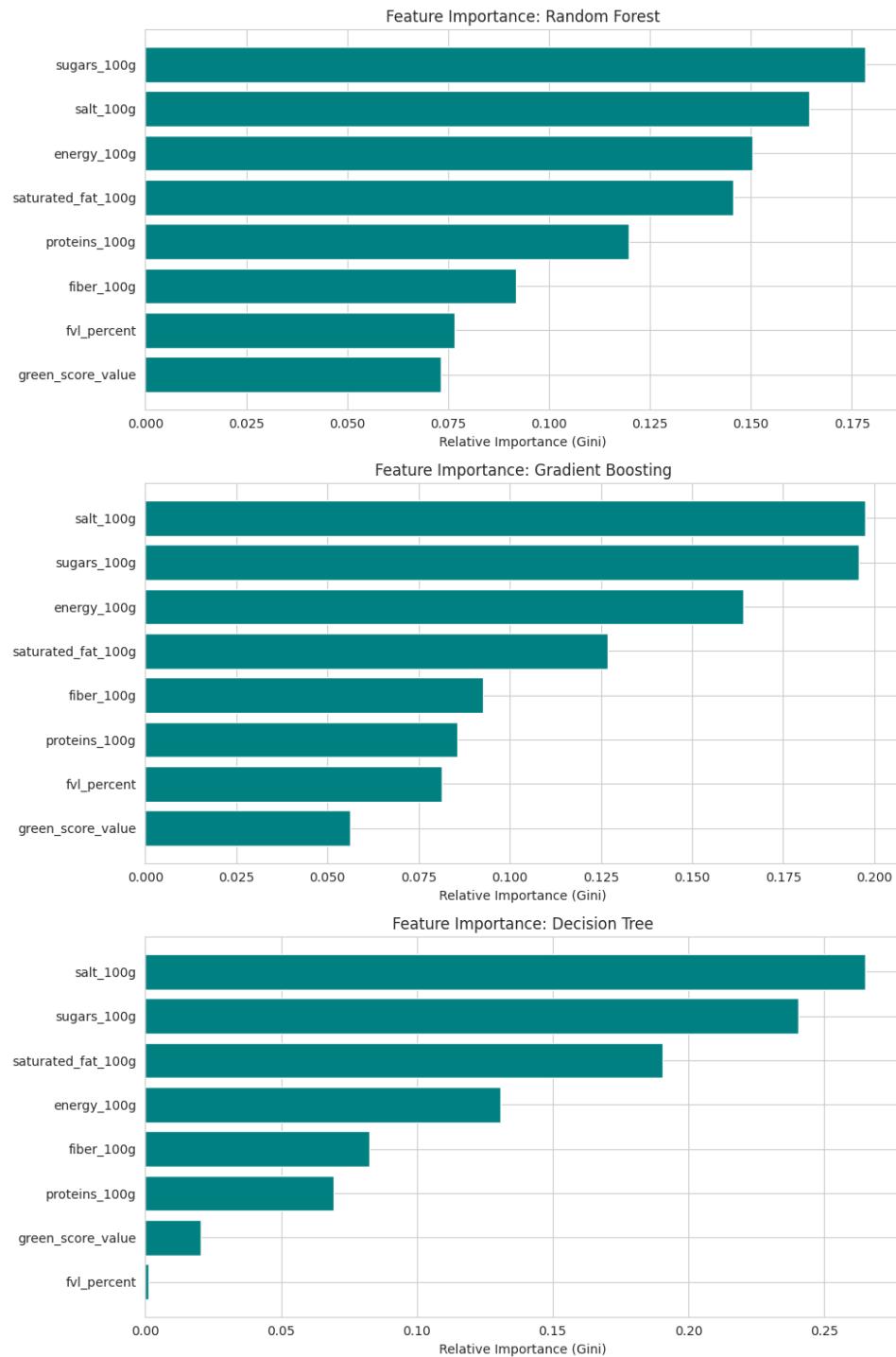


Figure 5.8: Feature Importance of Tree-based Models

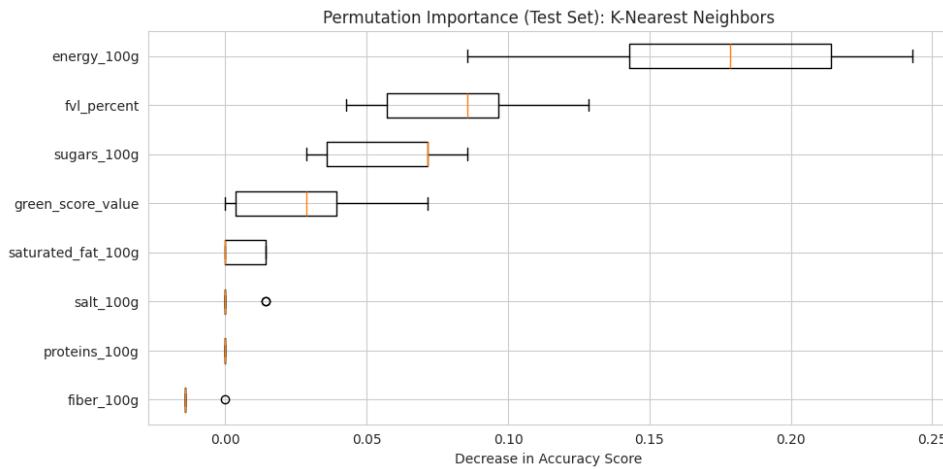


Figure 5.9: Permutation Importance of KNN Model

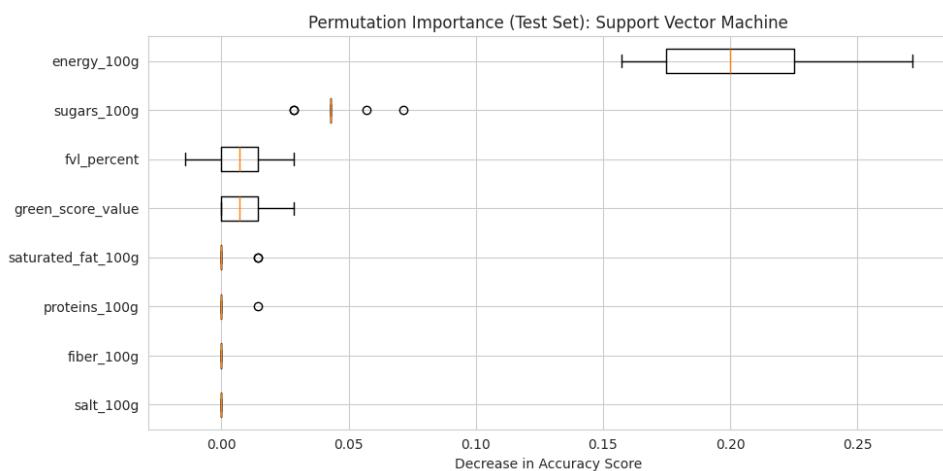


Figure 5.10: Permutation Importance of SVM Model

CHAPTER 6

Comparison with Other Group

The final stage of our project involved applying our decision models to a secondary dataset of 400 food products provided by another project group. It is important to emphasize that the goal of this comparison is **not to maximize accuracy relative to the official Nutri-Score**. Since our goal was to construct independent models with their own specific decision logic, we do not treat the official Nutri-Score as an absolute ground truth to be mimicked. Instead, this chapter evaluates the consistency and generalizability of our logic when faced with a new, unseen distribution of food products and compares our structural priorities with those of the other group.

The distribution of the data from the other group is shown in Figure 6.1. While our original dataset had a more balanced distribution between classes, this external dataset is skewed towards C and E classes, while the remaining classes are all lower and almost equal.

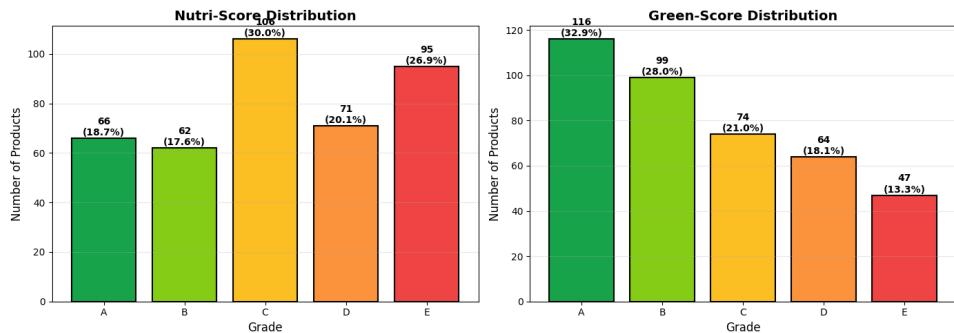


Figure 6.1: Distribution of Other Group's Dataset

6.1 Results and Analysis

For the other group, their most successful configuration was the optimistic version of the ELECTRE-TRI model using a majority threshold of $\lambda = 0.6$. As shown in Figure 6.2, their model achieved its peak similarity under these specific parameters.

We then ran the external dataset with one of our ELECTRE-TRI models to compare how our model classifies the products differently. We opted not to use our most similar model which uses differential optimization. Instead, we chose the pessimistic version of our model with equal weights and uniform profiles and a majority threshold of $\lambda = 0.6$. This is more interesting since it is closer to how the other group chose their weights and thresholds (i.e., they acted as decision-makers and chose specific quintiles for their profiles). The confusion matrix for this configuration is shown in Figure 6.3.

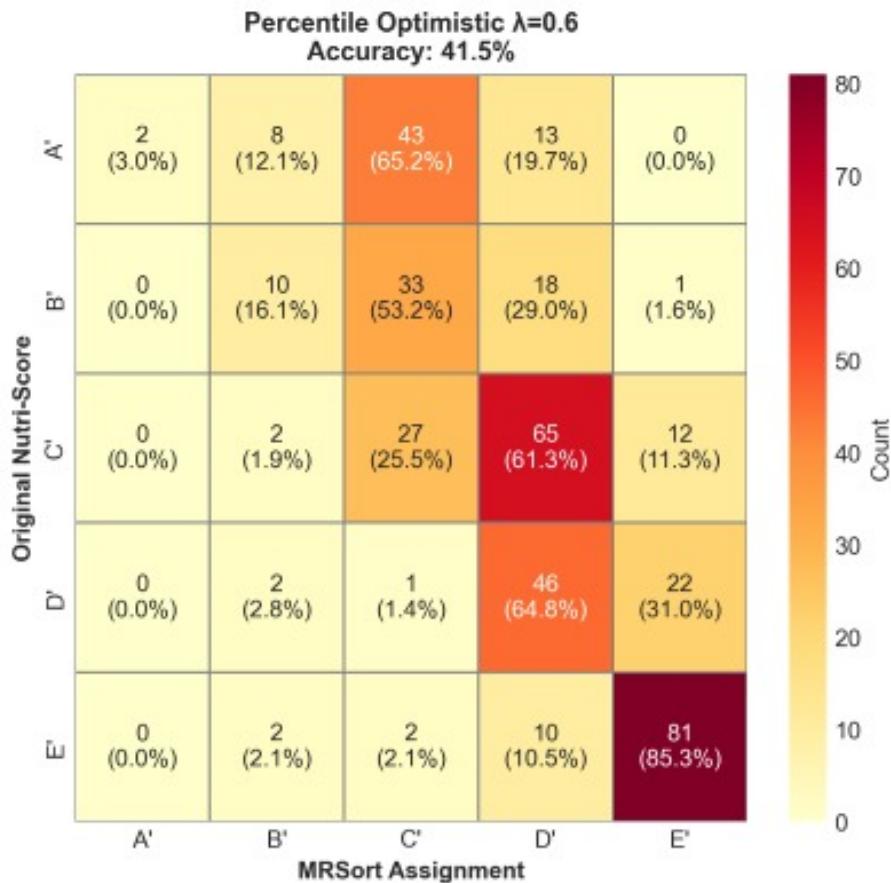


Figure 6.2: Confusion Matrix from other group on their data

When comparing these two confusion matrices, a clear structural difference is visible. The other group's optimistic model displays a distinct upward bias, where products are more frequently pushed toward higher categories like A and B, even when the official label is lower, which is as expected for optimistic models. In contrast, our pessimistic approach demonstrates a downward bias, often misclassifying below the diagonal. This indicates that while the other group's model is more forgiving, our model requires a much stronger consensus of positive nutrients to award a high grade. Furthermore, the other group's matrix shows more balanced activity in the middle classes (B, C, and D), whereas our uniform profile approach tends to polarize classifications toward the extremes (A and E). This suggests that their manually selected quintiles were more effective at capturing the non-linear boundaries of the official Nutri-Score than our strictly uniform intervals, which struggle to classify products in the middle ground. Our model is much more black-and-white in a sense, since it usually just puts products in either the healthiest or least healthy classes, while the other group's model balances classifications out between classes a lot more.

Another interesting experiment involved applying our optimized weighted sum model, which achieved our highest overall similarity to the Nutri-score, to the external dataset. As shown in Figure 6.4, the similarity to the official Nutri-Score dropped significantly to

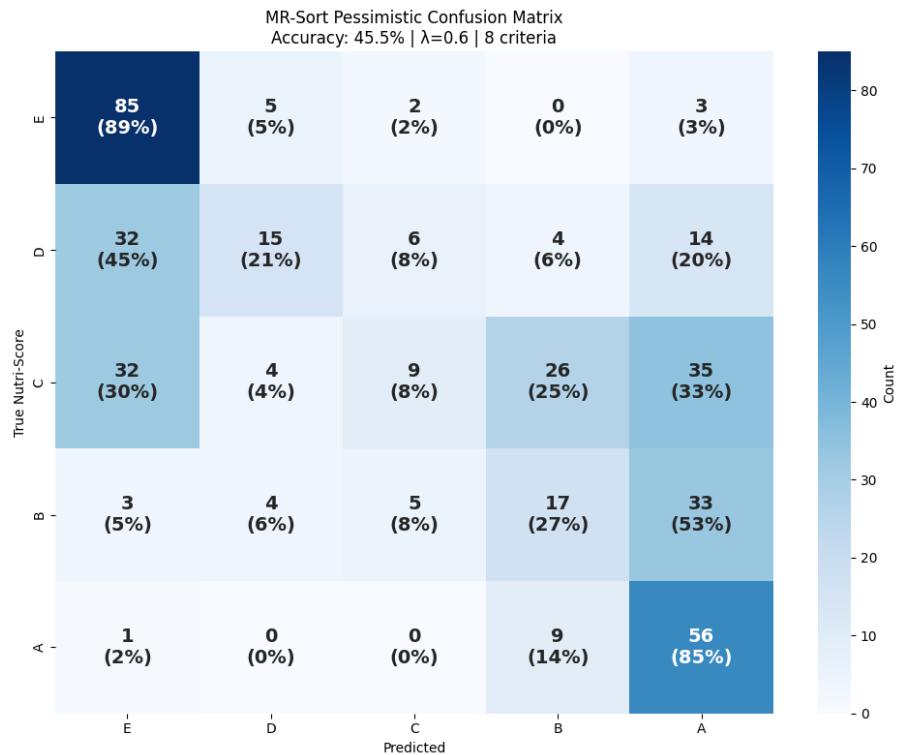


Figure 6.3: Confusion Matrix from our group on external data

51.2%. While this remains more accurate than a random guess, the heavy decline from our original results suggests that the differential evolution process may have overfitted to the specific nutritional distributions and boundaries of our primary dataset.

This outcome indicates that optimization may lack robustness when the model is developed on a relatively small subset of products. Despite the lower overall similarity, the model maintained 100% accuracy for Class E products, while performance for the remaining categories was not that good. This suggests that while the unhealthy end of the utility spectrum is consistent across datasets, the model's learned thresholds for higher classes are highly sensitive to the specific data used during training, which eventually ended up skewing the classifications towards lower classes.

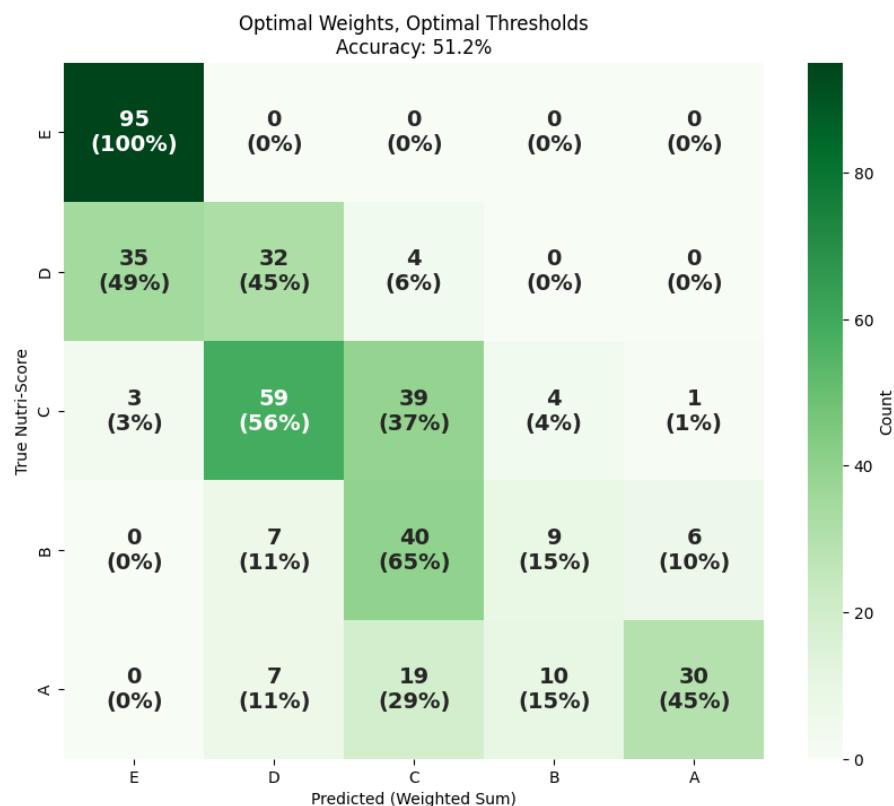


Figure 6.4: Confusion Matrix from our group using the optimized weighted sum model on external data

CHAPTER 7

Conclusion

This project examined the Nutri-Score system by using three different kinds of models: one based on rules and boundaries called ELECTRE-TRI, one based on adding up scores called the Weighted Sum Model, and one based on learning patterns from data called Machine Learning. We also included the Green-Score to see how environmental factors change the results.

7.1 Performance Summary

The models showed very different results. The Weighted Sum Model was the most accurate at matching the real Nutri-Score labels, reaching about 79% similarity. The Machine Learning models were also quite good, reaching around 75%. The ELECTRE-TRI model was the least similar to the official system, with its best result being around 50%.

7.2 The Model We Prefer

Among all the models we built, we feel most comfortable using the weighted sum model. First, the official Nutri-Score works by adding and subtracting points for different nutrients. The Weighted Sum Model works the same way, so it naturally fits the logic of the system we are trying to study. Second, unlike Machine Learning, this model is easy to explain. We can see exactly how much weight is given to salt or sugar, which is important for being fair and clear with consumers. Third, it is easy to change. We were able to add the Green-Score ourselves to see how sustainability affects the grade.

7.3 Final Thoughts

Even though the weighted sum model matched the labels best, ELECTRE-TRI is still useful because it has a natural veto. This means if a product has a very dangerous amount of salt, the model can automatically give it a bad grade regardless of its other nutrients. This feature is missing in the other models. Machine learning models are also promising, but it is highly dependent on the quality of the data that is being used, which can take some time to design.

APPENDIX A

Complete Confusion Matrices

This appendix provides the comprehensive set of confusion matrices for our different experiments. These matrices visualize the alignment between our models and the official Nutri-Score labels across different strategies.

A.1 Electre-TRI Model

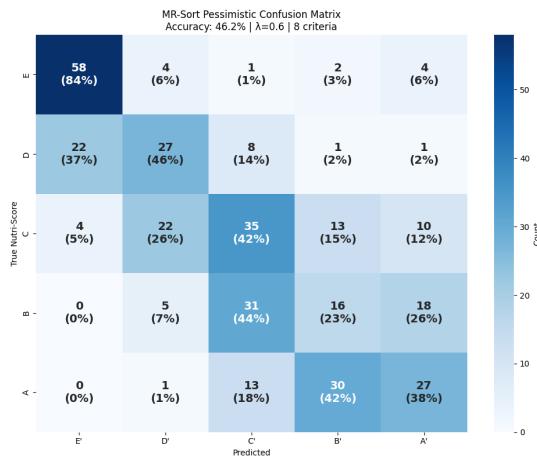


Figure A.1: Pessimistic ($\lambda = 0.6$)

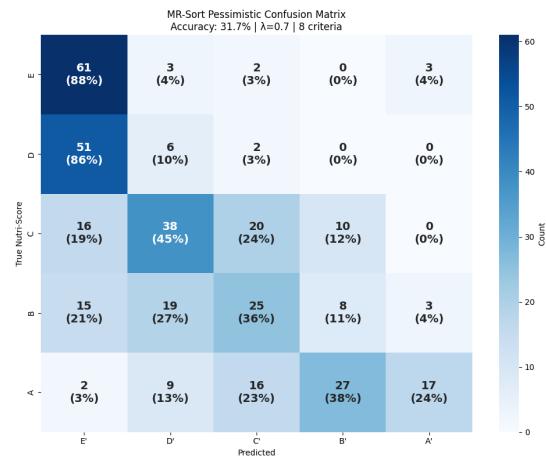


Figure A.2: Pessimistic ($\lambda = 0.7$)

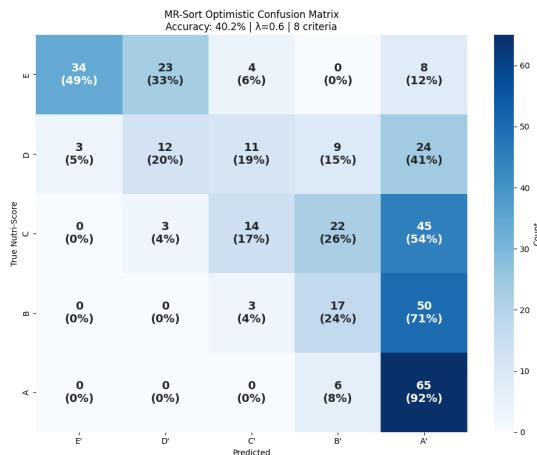


Figure A.3: Optimistic ($\lambda = 0.6$)

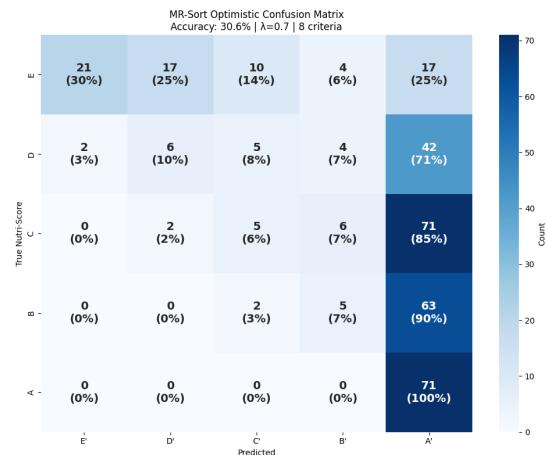


Figure A.4: Optimistic ($\lambda = 0.7$)

Figure A.5: Confusion matrices for **Correlation + Centroid models**.

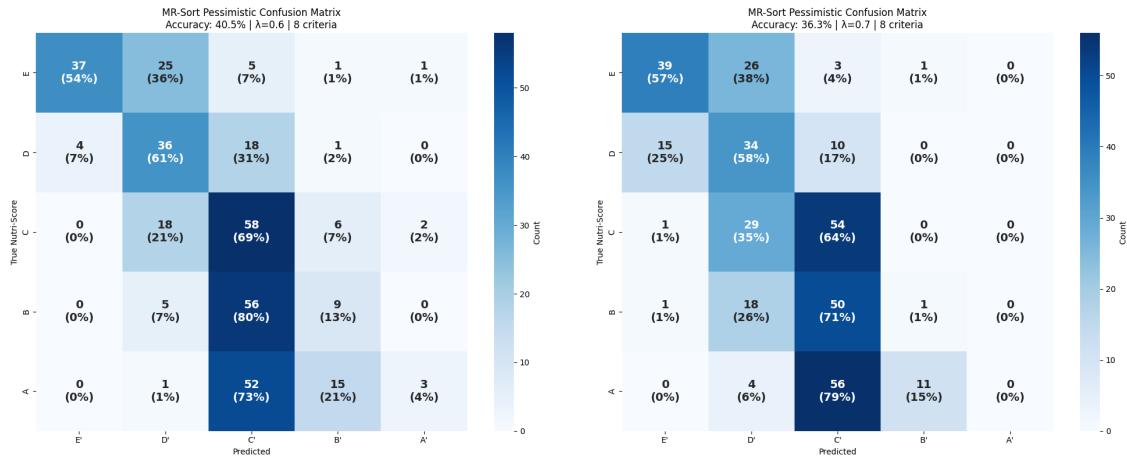
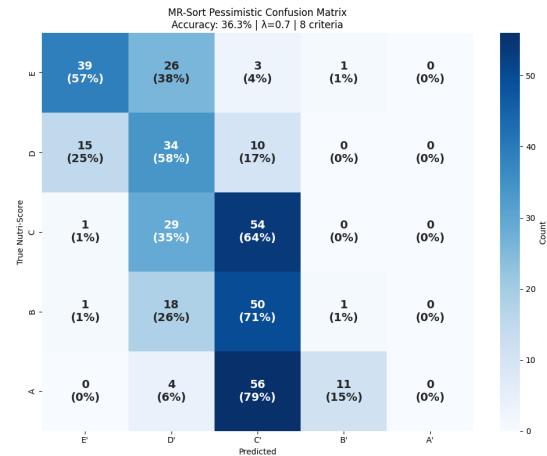
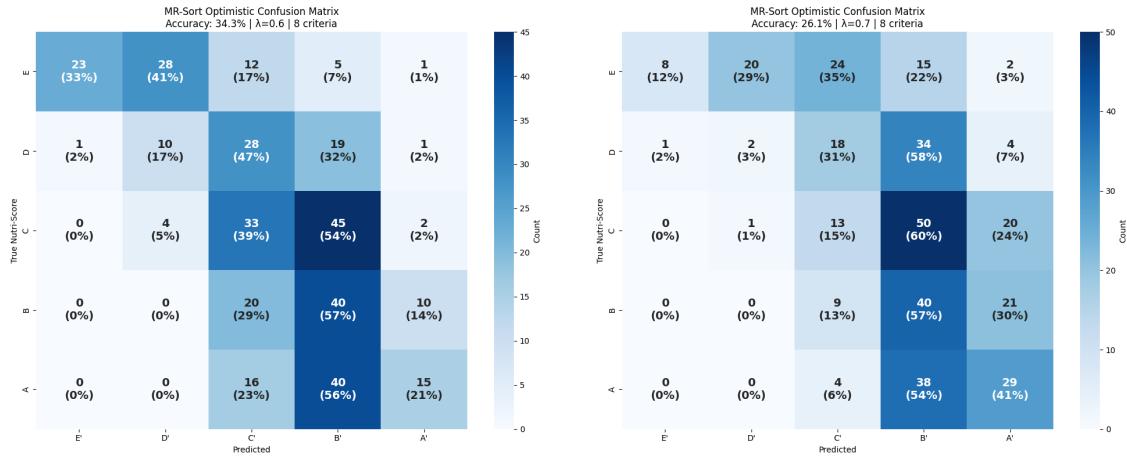
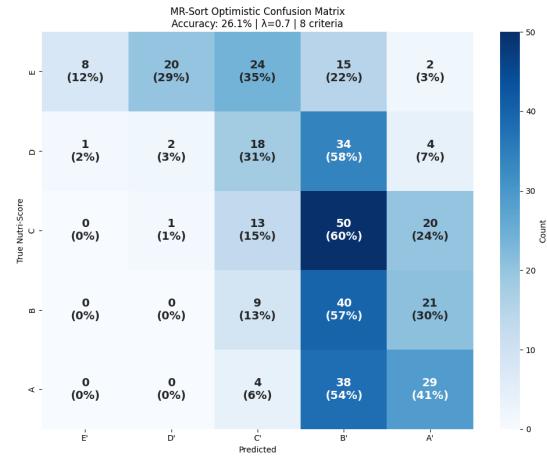

Figure A.6: Pessimistic ($\lambda = 0.6$)

Figure A.7: Pessimistic ($\lambda = 0.7$)

Figure A.8: Optimistic ($\lambda = 0.6$)

Figure A.9: Optimistic ($\lambda = 0.7$)

Figure A.10: Confusion matrices for Correlation + Distribution models.

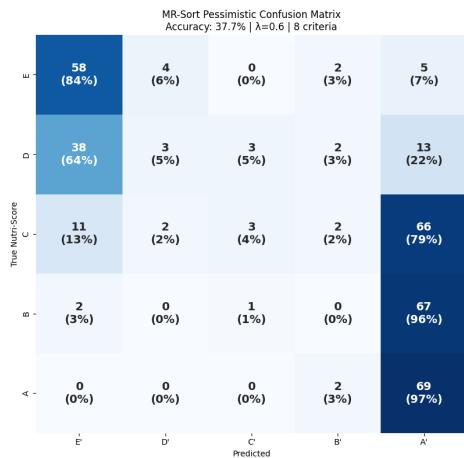
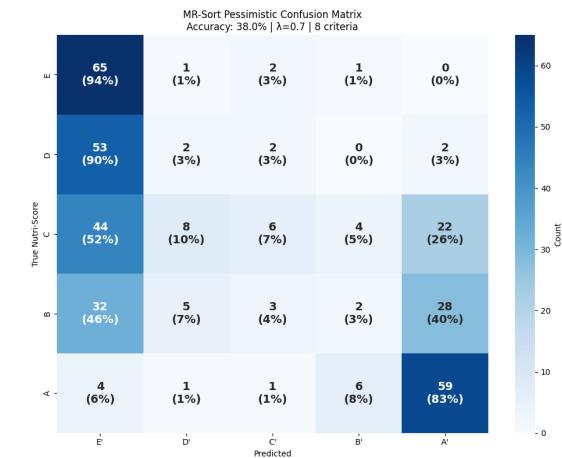
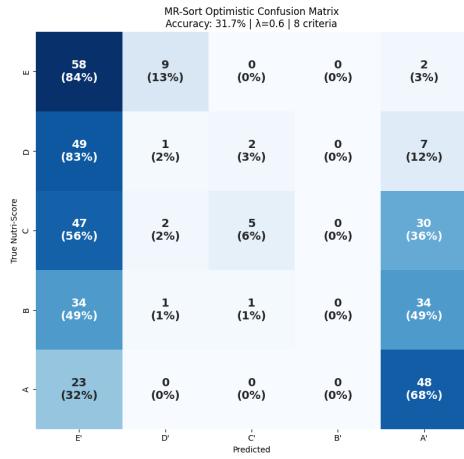
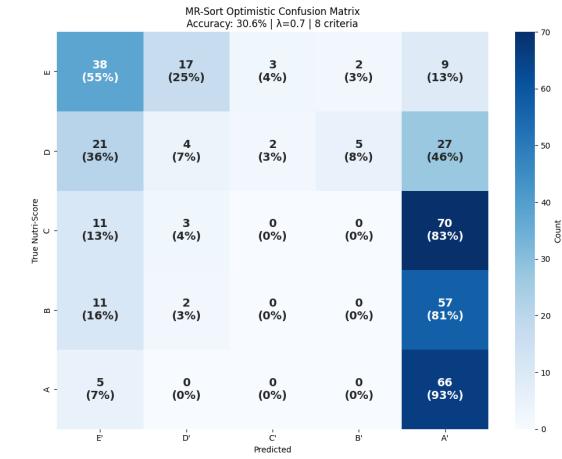
Figure A.11: Pessimistic ($\lambda = 0.6$)Figure A.12: Pessimistic ($\lambda = 0.7$)Figure A.13: Optimistic ($\lambda = 0.6$)Figure A.14: Optimistic ($\lambda = 0.7$)

Figure A.15: Confusion matrices for Correlation + Equal models.

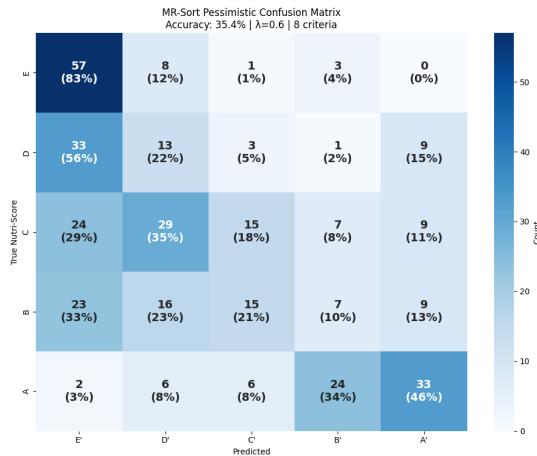
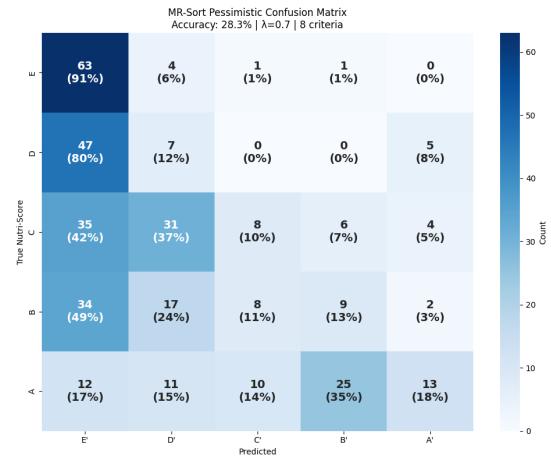
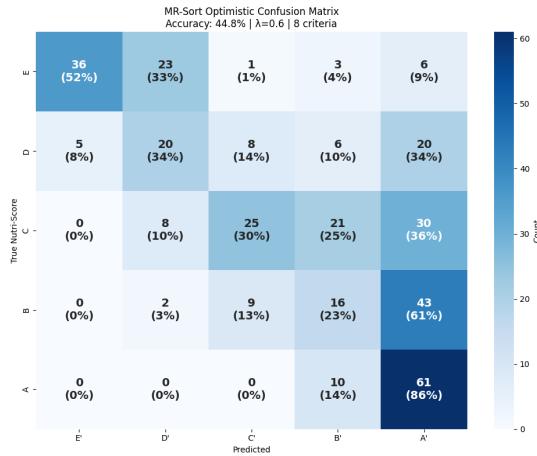
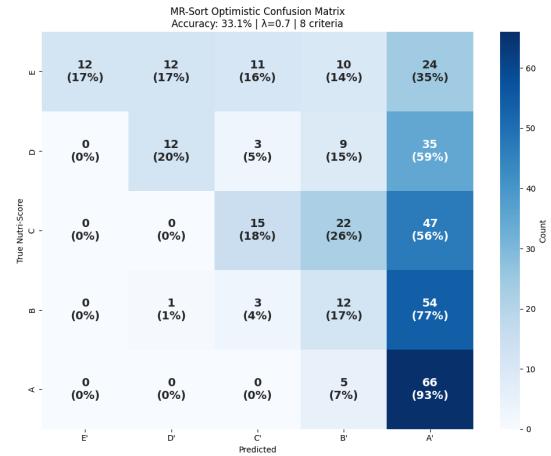

Figure A.16: Pessimistic ($\lambda = 0.6$)

Figure A.17: Pessimistic ($\lambda = 0.7$)

Figure A.18: Optimistic ($\lambda = 0.6$)

Figure A.19: Optimistic ($\lambda = 0.7$)

Figure A.20: Confusion matrices for Entropy + Centroid models.

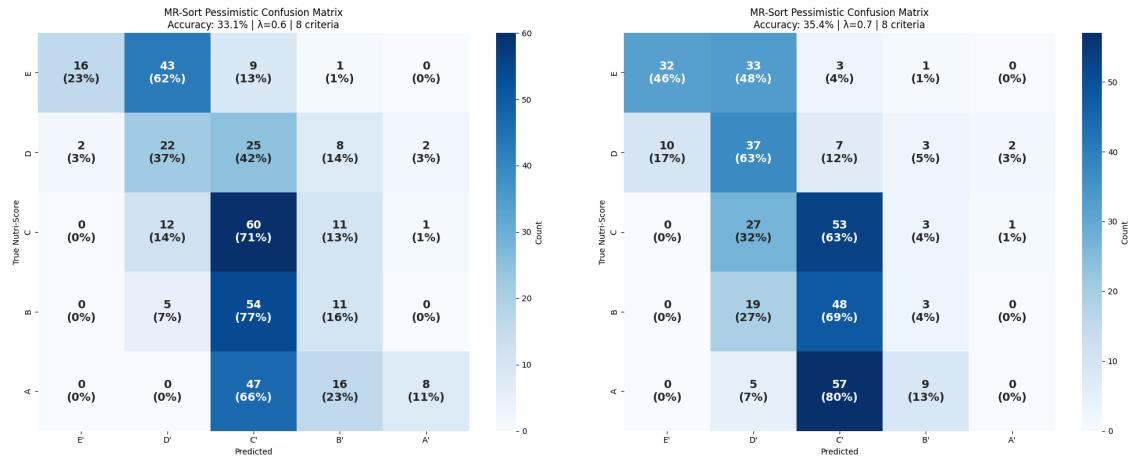
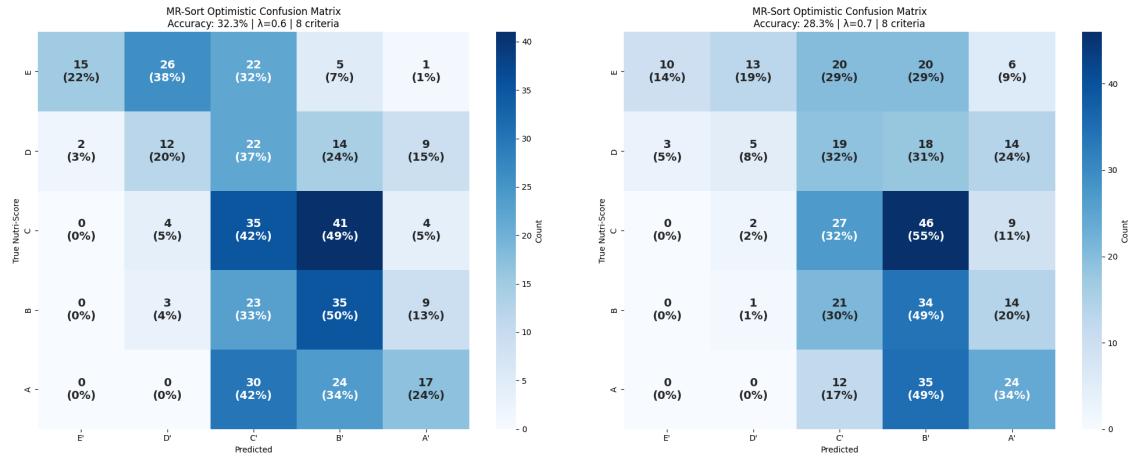
Figure A.21: Pessimistic ($\lambda = 0.6$)Figure A.22: Pessimistic ($\lambda = 0.7$)Figure A.23: Optimistic ($\lambda = 0.6$)Figure A.24: Optimistic ($\lambda = 0.7$)

Figure A.25: Confusion matrices for Entropy + Distribution models.

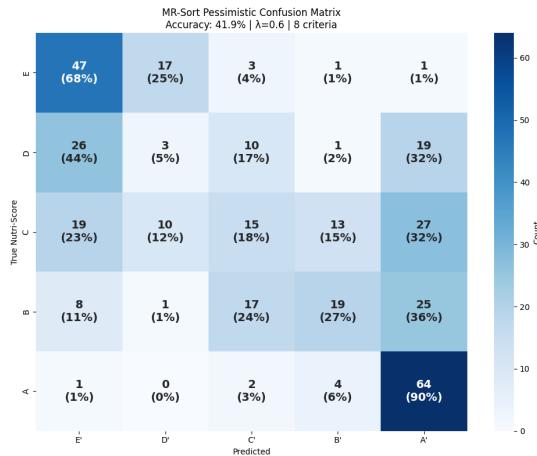
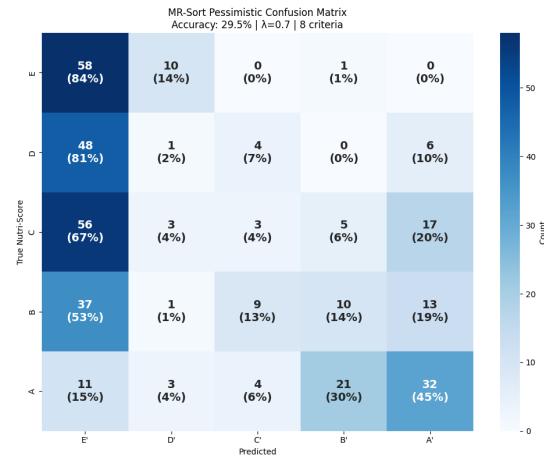

Figure A.26: Pessimistic ($\lambda = 0.6$)

Figure A.27: Pessimistic ($\lambda = 0.7$)

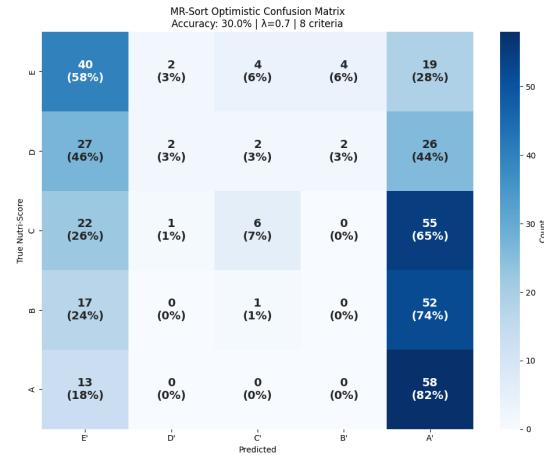
Figure A.28: Optimistic ($\lambda = 0.6$)

Figure A.29: Optimistic ($\lambda = 0.7$)

Figure A.30: Confusion matrices for Entropy + Equal models.

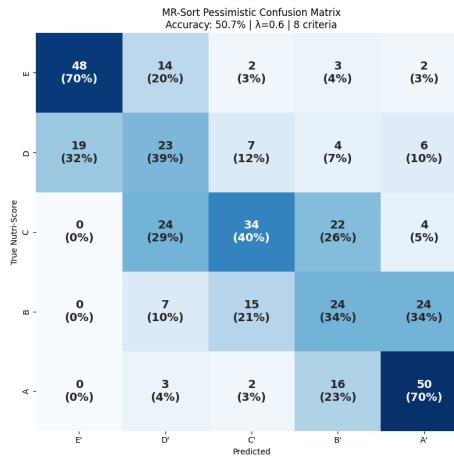
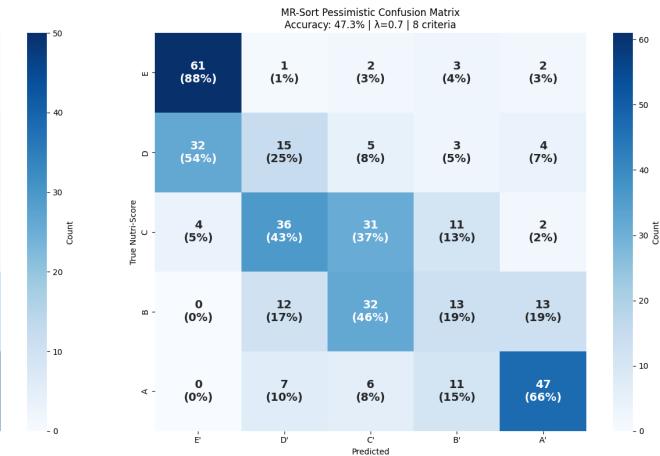
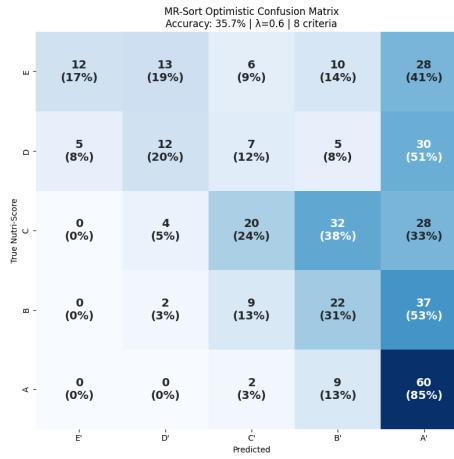
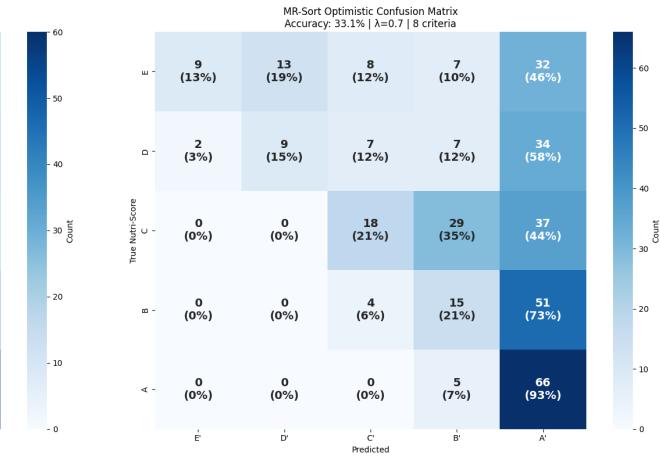
Figure A.31: Pessimistic ($\lambda = 0.6$)Figure A.32: Pessimistic ($\lambda = 0.7$)Figure A.33: Optimistic ($\lambda = 0.6$)Figure A.34: Optimistic ($\lambda = 0.7$)

Figure A.35: Confusion matrices for Optimized + Centroid models.

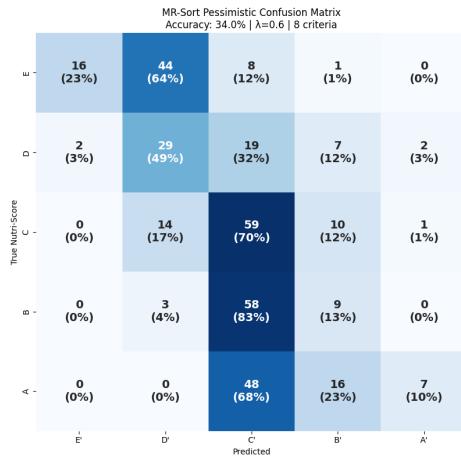
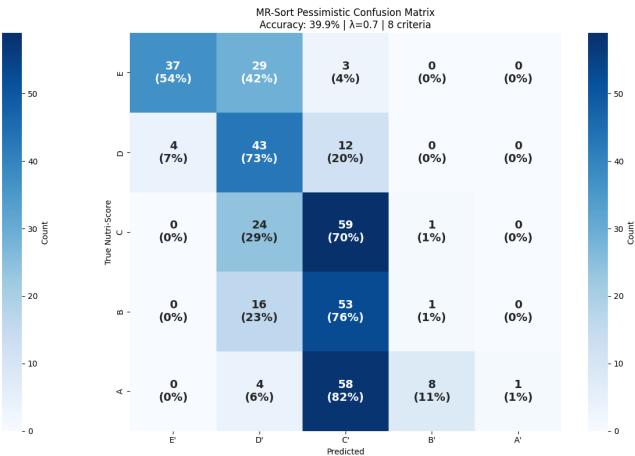
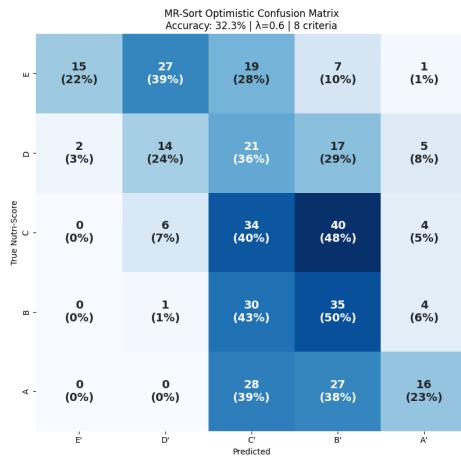
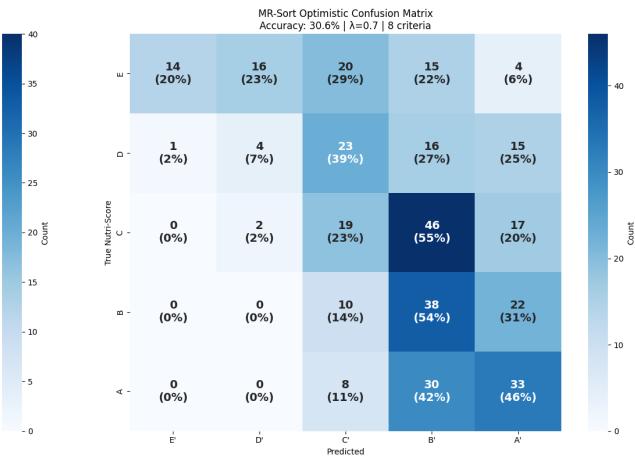

Figure A.36: Pessimistic ($\lambda = 0.6$)

Figure A.37: Pessimistic ($\lambda = 0.7$)

Figure A.38: Optimistic ($\lambda = 0.6$)

Figure A.39: Optimistic ($\lambda = 0.7$)

Figure A.40: Confusion matrices for Optimized + Distribution models.

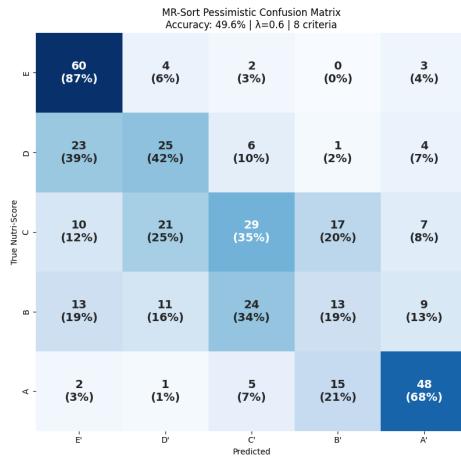
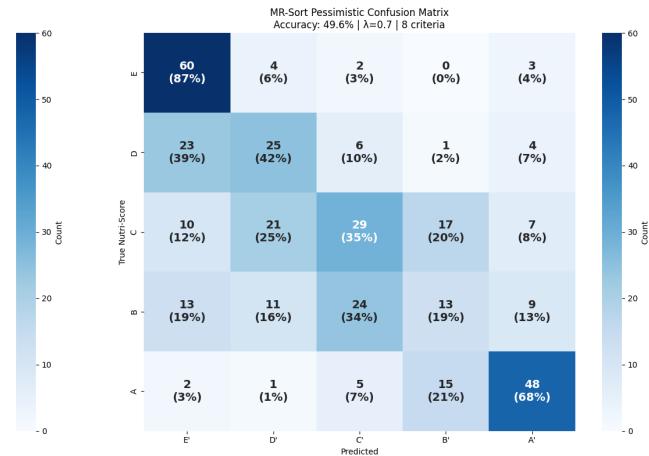
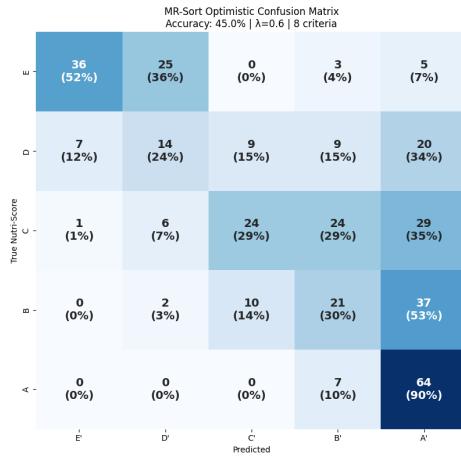
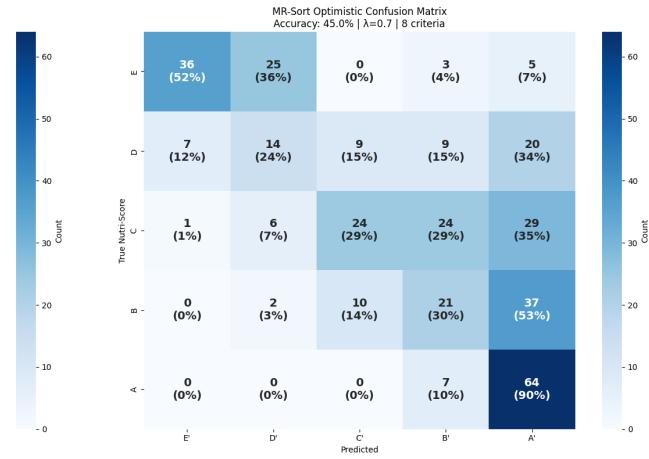
Figure A.41: Pessimistic ($\lambda = 0.6$)Figure A.42: Pessimistic ($\lambda = 0.7$)Figure A.43: Optimistic ($\lambda = 0.6$)Figure A.44: Optimistic ($\lambda = 0.7$)

Figure A.45: Confusion matrices for Equal + Centroid models.

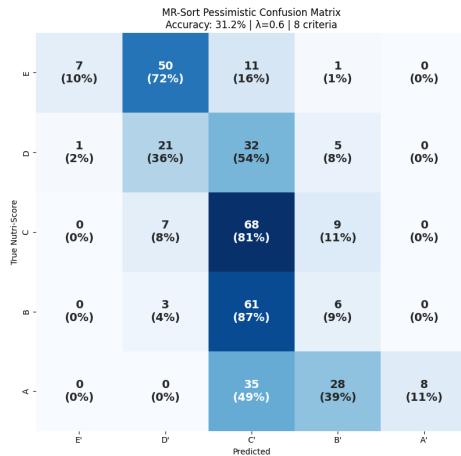
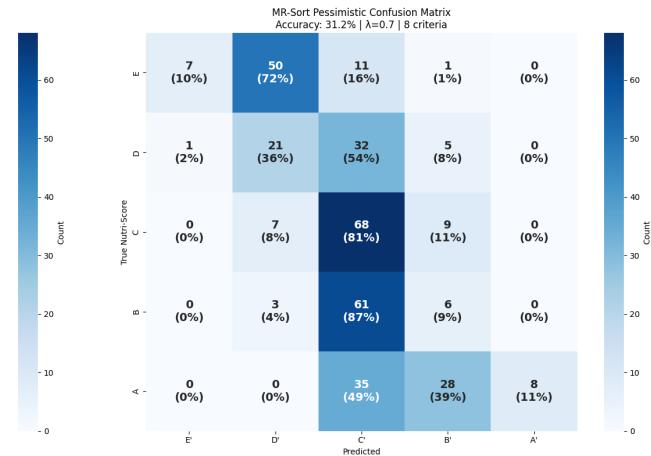
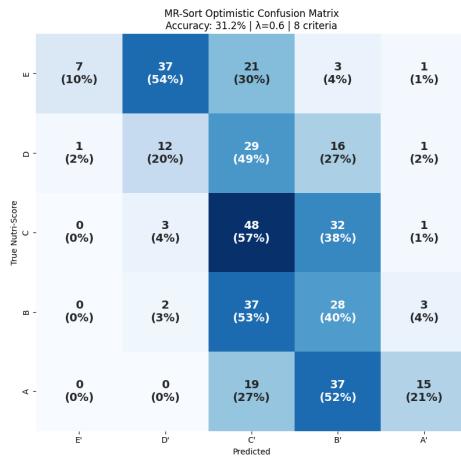
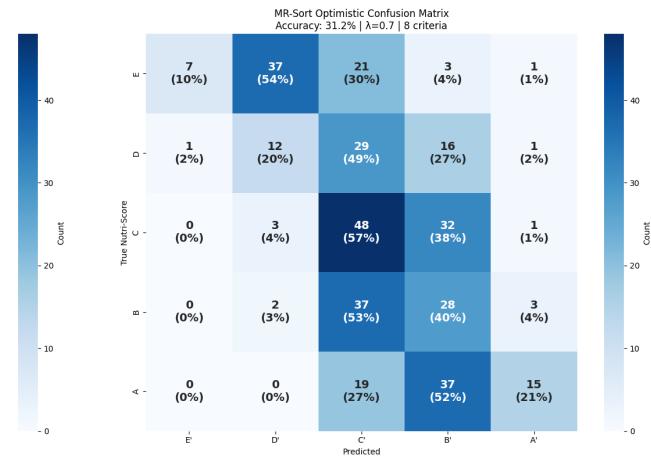

Figure A.46: Pessimistic ($\lambda = 0.6$)

Figure A.47: Pessimistic ($\lambda = 0.7$)

Figure A.48: Optimistic ($\lambda = 0.6$)

Figure A.49: Optimistic ($\lambda = 0.7$)

Figure A.50: Confusion matrices for Equal + Distribution models.

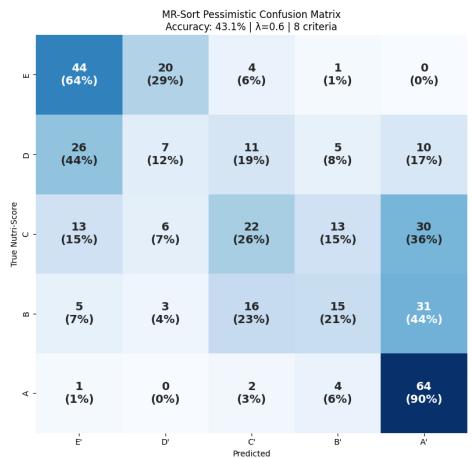
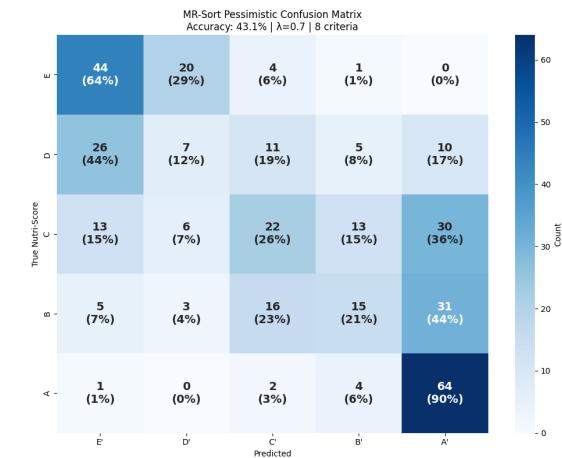
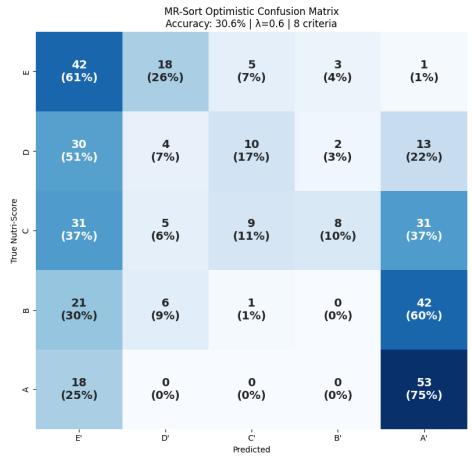
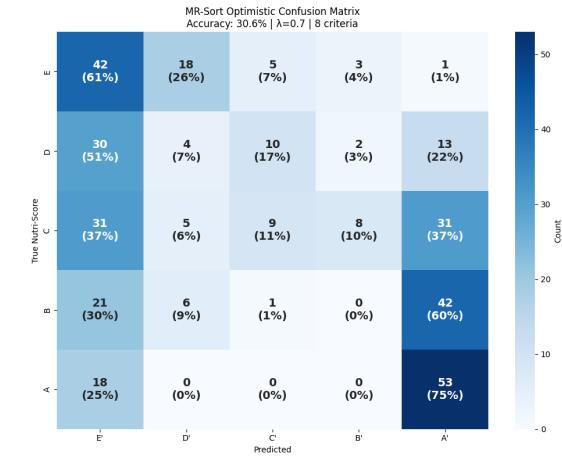
Figure A.51: Pessimistic ($\lambda = 0.6$)Figure A.52: Pessimistic ($\lambda = 0.7$)Figure A.53: Optimistic ($\lambda = 0.6$)Figure A.54: Optimistic ($\lambda = 0.7$)

Figure A.55: Confusion matrices for Equal + Equal models.

A.2 Weighted Sum Model

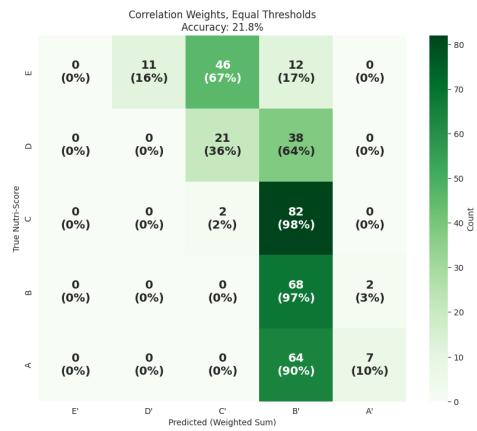


Figure A.56: Correlation Weights + Equal.

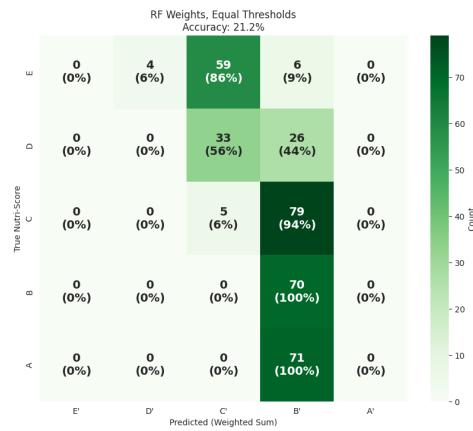


Figure A.57: Random Forest Weights + Equal.

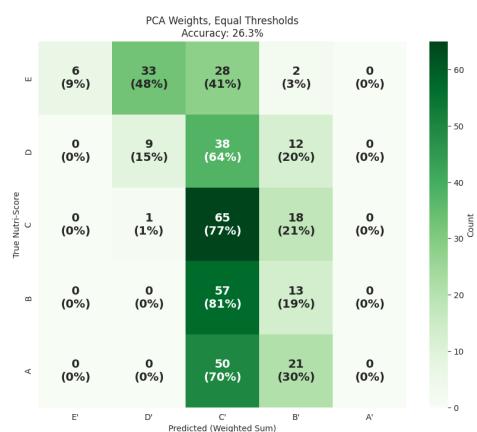


Figure A.58: PCA Weights + Equal.

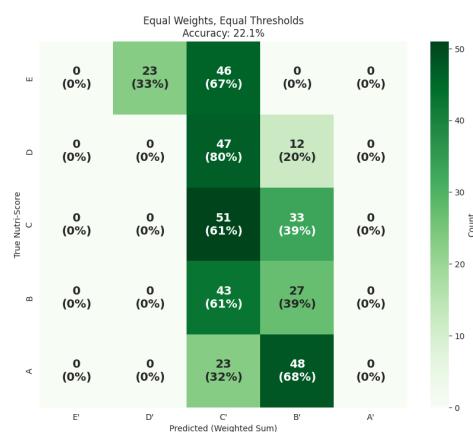


Figure A.59: Equal Weights + Equal.

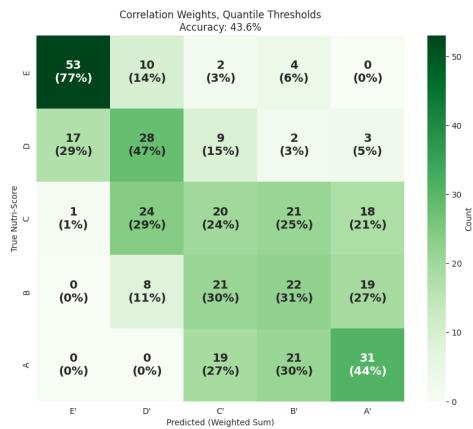


Figure A.60: Correlation Weights + Quantile.

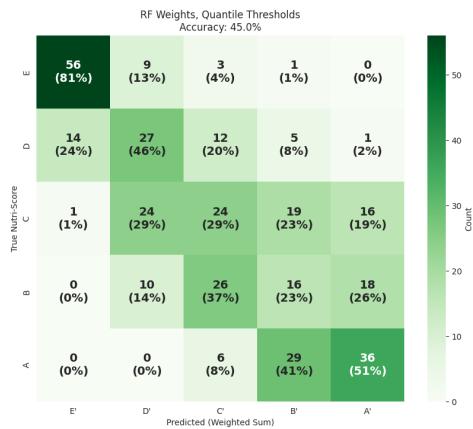


Figure A.61: Random Forest Weights + Quantile.

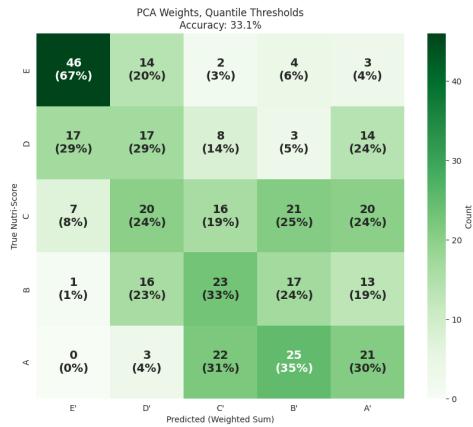


Figure A.62: PCA Weights + Quantile.



Figure A.63: Equal Weights + Quantile.

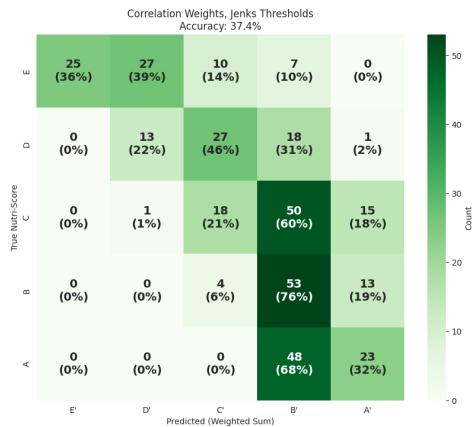


Figure A.64: Correlation Weights + Jenks.

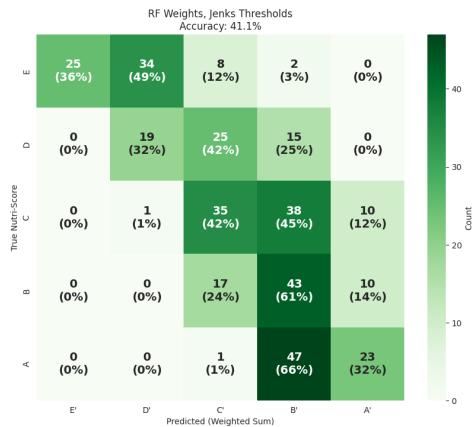


Figure A.65: Random Forest Weights + Jenks.

A.2. Weighted Sum Model

55

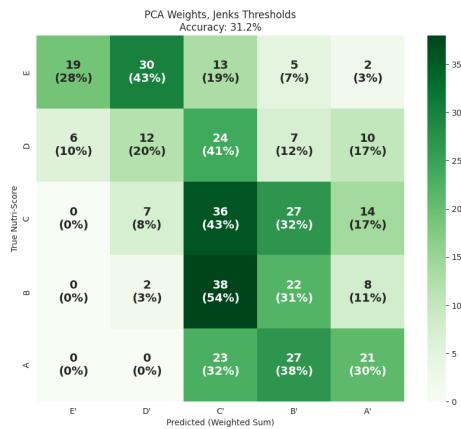


Figure A.66: PCA Weights + Jenks.

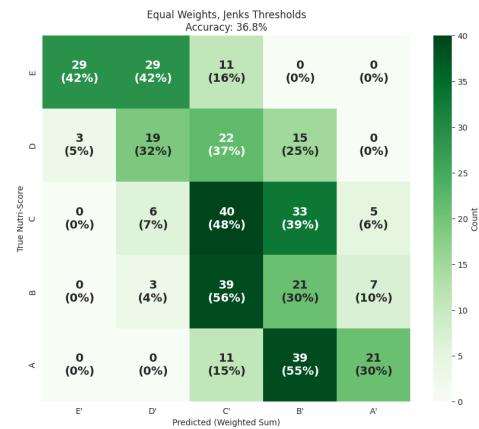


Figure A.67: Equal Weights + Jenks.

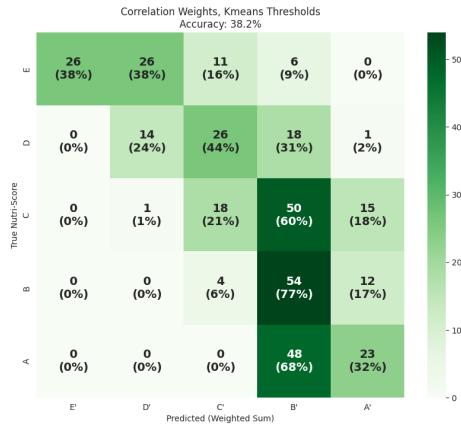


Figure A.68: Correlation Weights + Kmeans.

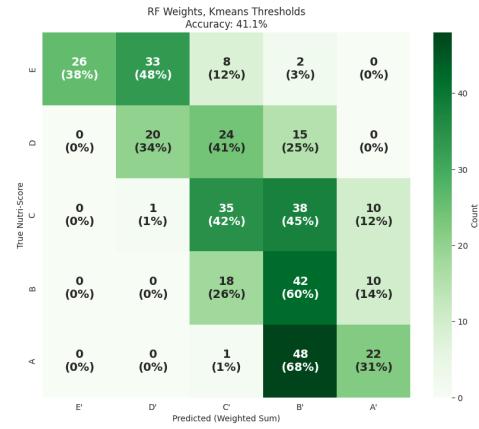


Figure A.69: Random Forest Weights + Kmeans.

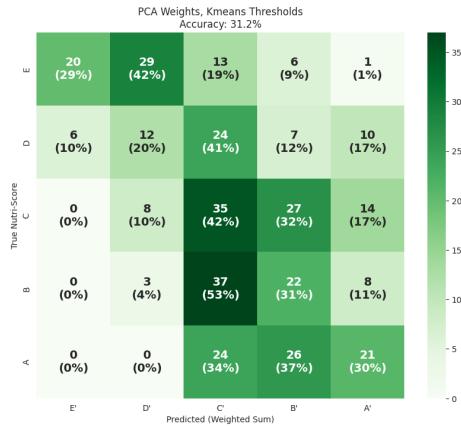


Figure A.70: PCA Weights + Kmeans.

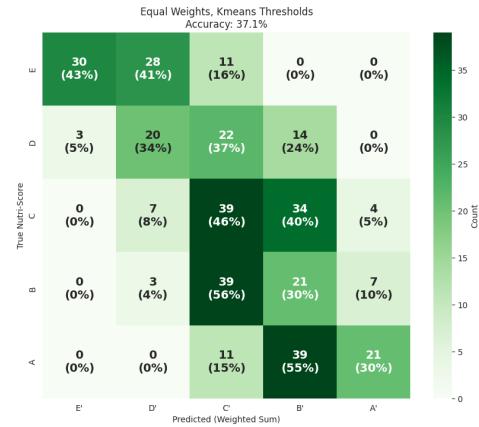


Figure A.71: Equal Weights + Kmeans.

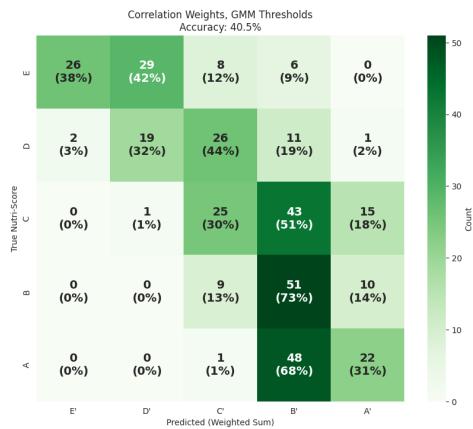


Figure A.72: Correlation Weights + GMM.

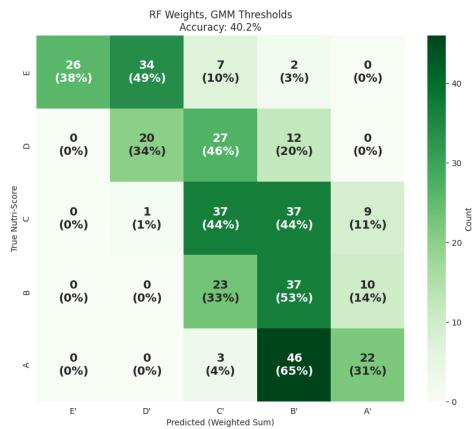


Figure A.73: Random Forest Weights + GMM.

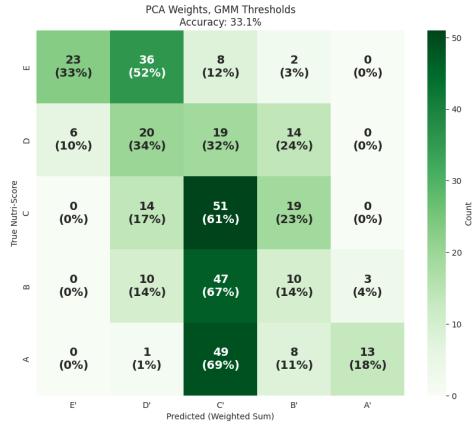


Figure A.74: PCA Weights + GMM.

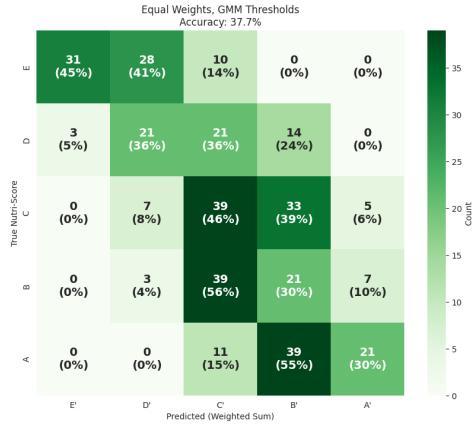


Figure A.75: Equal Weights + GMM.

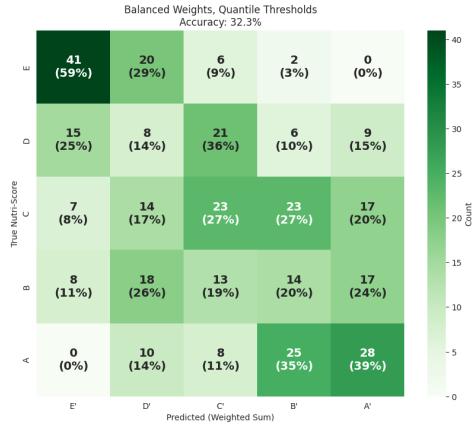


Figure A.76: Balanced, Sustainable Weights + Quantile.

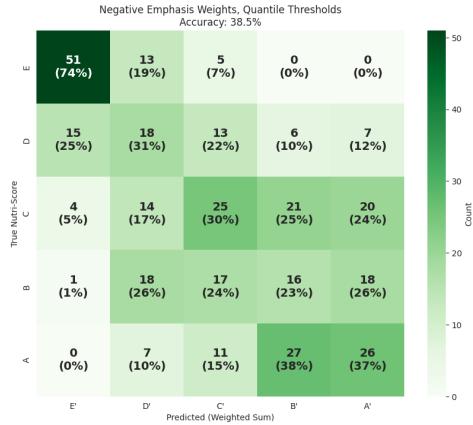


Figure A.77: Negative-Heavy Weights + Quantile.

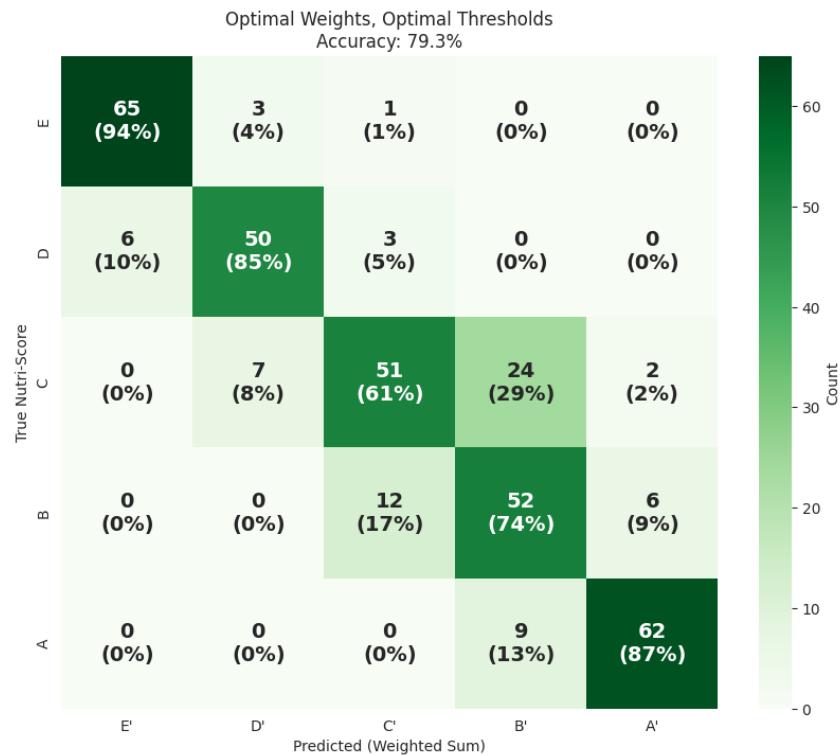


Figure A.78: Optimized

Bibliography

- [1] CentraleSupélec. Elaboration of some decision models for the nutri-score label. Master BDMA Decision Modeling Project Statement, 2025. Course Material. (Cited on pages 3 and 4.)
- [2] Open Food Facts Contributors. Open food facts api documentation, 2024. (Cited on pages 1 and 3.)