Decision Modeling Project (DM)

Erasmus Mundus Joint Master's Degree
in
Big Data Management and Analytics

by

Ali ABUSALEH
Rishika GUPTA

under the guidance of

Prof. Brice Mayag

Informatique et Sciences du Numérique
CentraleSupélec (Université Paris-Saclay)
December 2023

Contents

List of Figures

List of Tables

Chapter 1

Database

1.1 Dataset Overview

The Open Food Facts API [1] serves as a vast repository of food product information, capturing an extensive array of details ranging from nutritional content to ingredients and packaging. The database chosen for the project is a small subset of

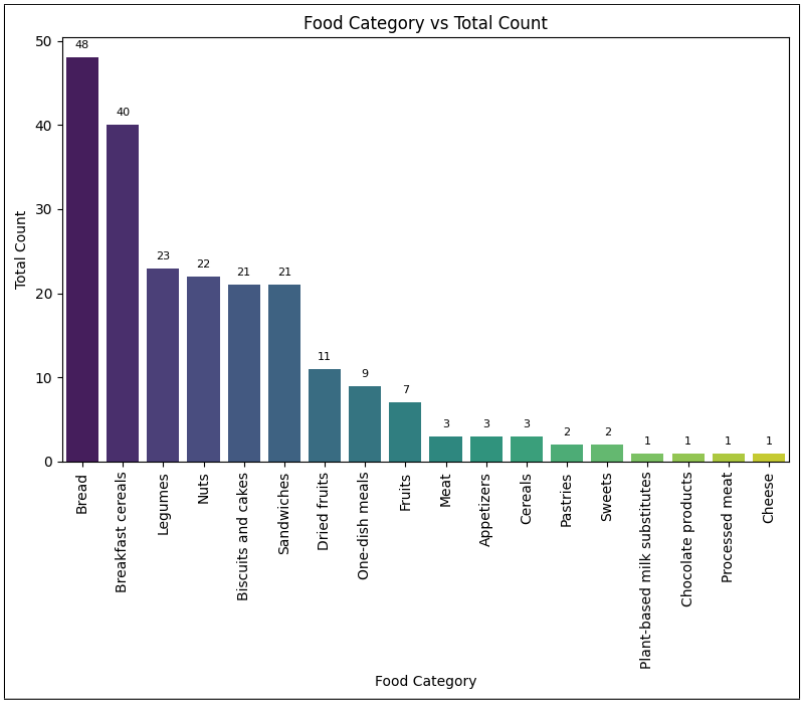

Figure 1.1: Food categories vs total count

the actual data, which is for the products from **all over the world**, not any specific region or country. Furthermore, the database draws information from various food products, categorizing them based on multiple categories - **Biscuits, Breads, Nuts, Sandwiches, Snacks, Meat alternatives, Chocolate candies, Breakfast cereals, and Fruits**. The visualization depicted in Figure 1.1 illustrates the aggregated count distribution across different categories and their corresponding subcategories. Each product entry is structured with a comprehensive set of fields encompassing nutritional data, ingredient lists, product labels, and more.

The Nutriscore calculation involves a complex algorithm considering diverse nutritional parameters. Elements such as energy content, saturated fats, sugars, sodium, protein, fiber, and the presence of fruits/vegetables/nuts contribute to the score. So, before proceeding any further, some exclusions are made:

- Beverage categories have been excluded to streamline analysis.
- Products with negative points surpassing the threshold of 11 have been omitted, indicating a deviation from desired nutritional standards as per the Nutriscore equation [2].

1.2 Data Preprocessing

The data preprocessing phase began by selecting critical columns, including **_id, image_url, brands, pnns_groups_2, nutriments, nutriscore_data, and nutrition_grade_fr**, crucial for product identification, brand information, and nutritional analysis.

To ensure data integrity, rows with missing nutritional information within the nutriments column were identified and subsequently dropped from the dataset, allowing for a more comprehensive and accurate analysis. The complex nutriments field, stored in JSON format, underwent restructuring by flattening it into a more structured form, facilitating detailed nutritional analysis and interpretation. Simultaneously, the nutriscore_data was validated for accuracy and consistency to ensure reliability in Nutriscore-related information. Following data refinement, the dataset

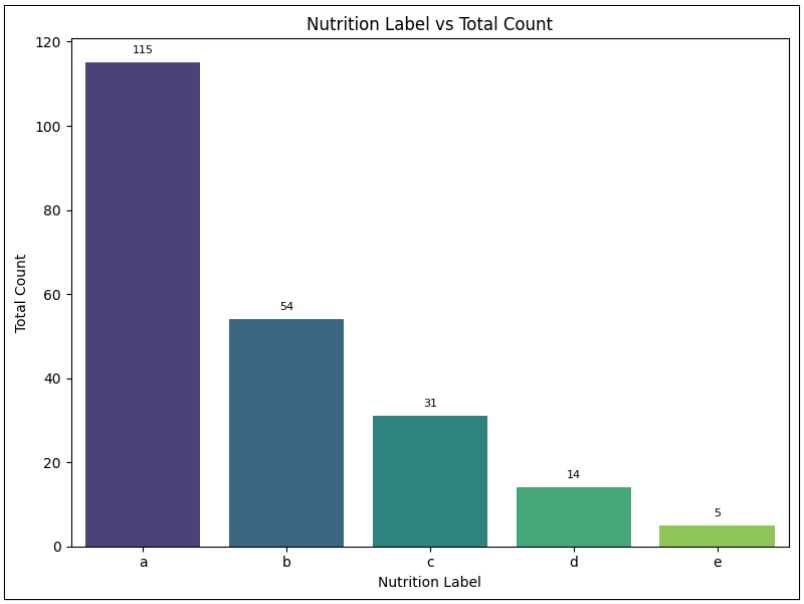

Figure 1.2: Nutritional categories vs total count

was filtered to retain only the most relevant columns essential for the planned analyses. From the filtered dataset, a final preprocessed subset comprising 300 randomly sampled items was created, serving as a representative subset for subsequent in-depth analyses while managing computational complexity. Figure 1.2 showcases the distribution of nutritional labels (denoted as a, b, c, d and e) within the selected data samples.

Chapter 2

Additive Model

2.1 Study Correlations

Understanding and addressing correlations among nutritional components are pivotal in constructing an accurate additive model for nutritional assessment. By acknowledging interdependencies and employing appropriate analytical strategies, the model's robustness can be enhanced, ensuring a more accurate evaluation of nutritional content and its impact on dietary assessment and health evaluation. This section delves into understanding and addressing these correlations to construct a robust assessment model.

$$\begin{aligned} \text{Energy} = (9 \times \text{Fat}) + (7 \times \text{Alcohol}) + (4 \times \text{Protein}) \\ + (4 \times \text{Sugar}) + (2.4 \times \text{Organic acid}) + (2 \times \text{Fibers}) \end{aligned} \tag{2.1}$$

The physical equation 2.1 proposed for nutritional assessment integrates essential nutritional elements: fat, alcohol, protein, sugars, organic acids, and fibers. However, **due to missing elements such as alcohol and organic acid, the complete exclusion of energy is not fair**. Figure 2.1 illustrates the correlation heatmap

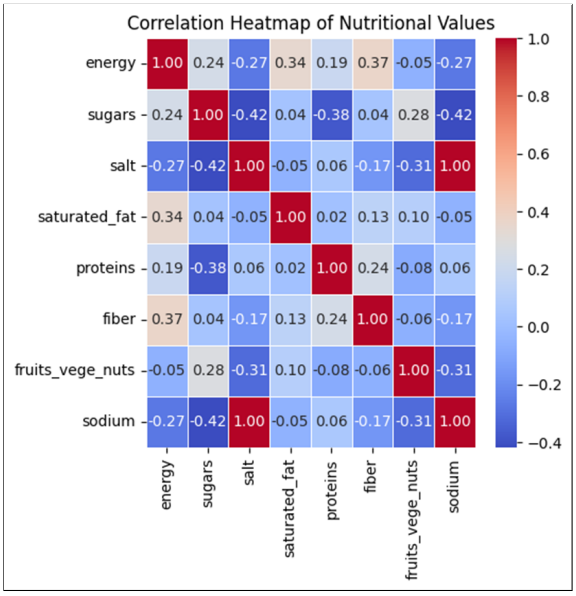

Figure 2.1: Correlation heatmap of nutritional values

of nutritional values. From the generated heatmap, several notable observations are discernible:

- **Salt/Sodium Correlation**: A significant correlation between salt and sodium components, showcases a complete correlation of 100%. This high correlation implies potential redundancy in retaining both elements, allowing for the exclusion of one while preserving essential nutritional information.
- **Energy-Fat-Fiber Correlation**: A substantial correlation between energy, fat, and fiber signifies a notable interdependence, suggesting that variations in one component might influence the others significantly.
- **Negative Correlation with Fruits**: A distinctive observation is the negative correlation observed between energy and fruits. This negative correlation implies a potential inverse relationship or contrasting impacts between energy intake and fruit consumption, hinting at varying dietary patterns or contrasting influences on overall nutritional composition.

2.2 Determining Weights

In this section, we delve deeper into the methodology used to assign the weights for our additive model. The initial step involved analyzing correlations among various

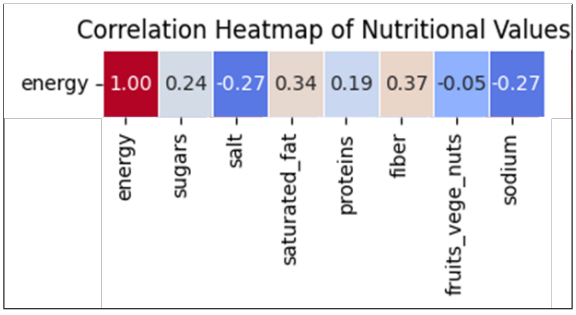

Figure 2.2: Additive Model: Correlation matrix to identify weights

nutritional values, as depicted in Figure 2.2.

- Notable observations as discussed in the previous section include the significant correlation of energy and sugars with fats and fibers. This led to the decision to incorporate 50% of energy (weight = 0.5) in the model while retaining fats (weight = 0.4) and fibers (weight = 0.3) in their original proportions due to their strong associations.
- Attributes like sugars, proteins and salt exhibited weaker correlations with other components. As a result, their weights were maintained at their original values: sugars at 10% (weight = 0.1), proteins at 20% (weight = 0.2) and salt at 35% (weight = 0.35).
- The presence of a negative relationship between fibers and energy necessitated a unique approach. To account for this inverse correlation, the weight assigned to fibers was adjusted to -5% (weight = -0.05), reflecting their opposing impacts within the model.

- Other correlations between the others were also taken into account and the weights were adjusted accordingly.

The weight distribution allocated within our additive model is detailed in Table 2.1.

Nutrition	**Energy**	**Sugars**	**Fats**	**Salt**	**Proteins**	**Fiber**	**Fruits**
Weight	0.5	0.1	0.4	0.35	0.2	0.3	-0.05

Table 2.1: Additive Model: Our assigned weights

Points	Fruits Légumes Fruits à coque (%)	Fibres (g/100g)		Protéines g/100g)
		NSP	ou AOAC	
0	≤40	≤0,7	≤0,9	≤1,6
1	>40	>0,7	>0,9	>1,6
2	>60	>1,4	>1,9	>3,2
3	-	>2,1	>2,8	>4,8
4	-	>2,8	>3,7	>6,4
5	>80	>3,5	>4,7	>8,0

Points	Valeur énergétique (kJ/100g)	Acides gras saturés (g/100g)	Sucres (g/100g)	Sodium (mg/100g)
0	≤335	≤1	≤4,5	≤90
1	>335	>1	>4,5	>90
2	>670	>2	>9	>180
3	>1005	>3	>13,5	>270
4	>1340	>4	>18	>360
5	>1675	>5	>22,5	>450
6	>2010	>6	>27	>540
7	>2345	>7	>31	>630
8	>2680	>8	>36	>720
9	>3015	>9	>40	>810
10	>3350	>10	>45	>900

Figure 2.3: Additive Model: Marginal utility functions [3]

Next, each nutritional value's weight was multiplied by its corresponding marginal utility function (f(x)) as elaborated in 2.3 to derive partial scores. This multiplication allowed for the consideration of each component's importance within the model and its impact on the final aggregated score. Also, an important constraint was identified through the marginal utility function, indicating a maximum achievable score of 15.75. This upper limit was integrated into the model, ensuring that the final score remained within this predefined boundary.

$$\begin{aligned} P(x) = 15.75 &- (0.5 \times f(Energy)) + (0.1 \times f(Sugar)) + (0.4 \times f(Fat)) \\ &+ (0.35 \times f(Salt)) + (0.2 \times f(Proteins)) + (0.3 \times f(Fibers)) \\ &- (0.05 \times f(Fruits)) \end{aligned} \tag{2.2}$$

The culmination of these considerations and calculations is represented by the

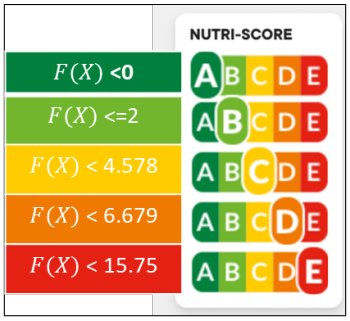

Figure 2.4: Additive Model: Nutriscore label [4]

additive model equation 2.2. By establishing Grade $G(P(x))$ as shown in figure 2.4, we map the assigned points $P(x)$ to a corresponding nutritional grade, which involves assigning grades based on the points accrued from the specific nutritional parameter within the context of the entire nutritional spectrum, which compensates for the small fractions ignored earlier.

2.3 Additive Model Results and Analysis

Figure 2.5 showcases our database foods categorized into nutritional labels using the developed additive model. This visualization highlights our model's comparison with the original labels. While our model appears to be more stringent in classifying

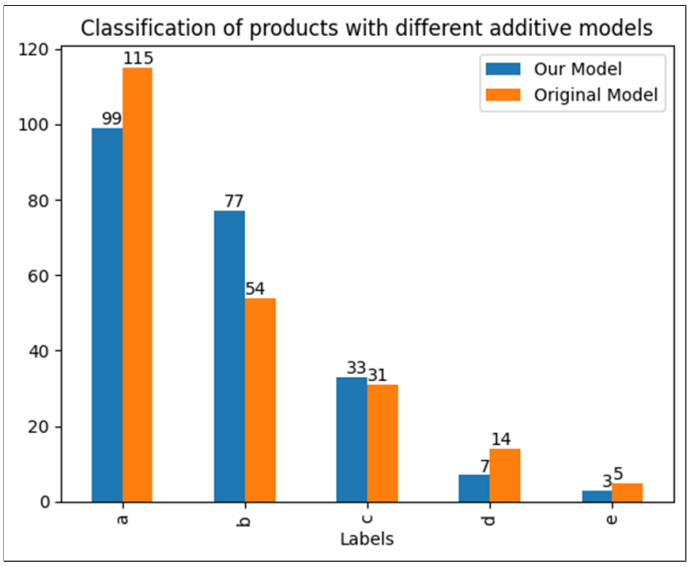

Figure 2.5: Additive Model: Comparison with original label

items into the a class, it's essential to note that the focus isn't solely on counts. To gain a deeper insight, we visualized the confusion matrix (Figure 2.6). The

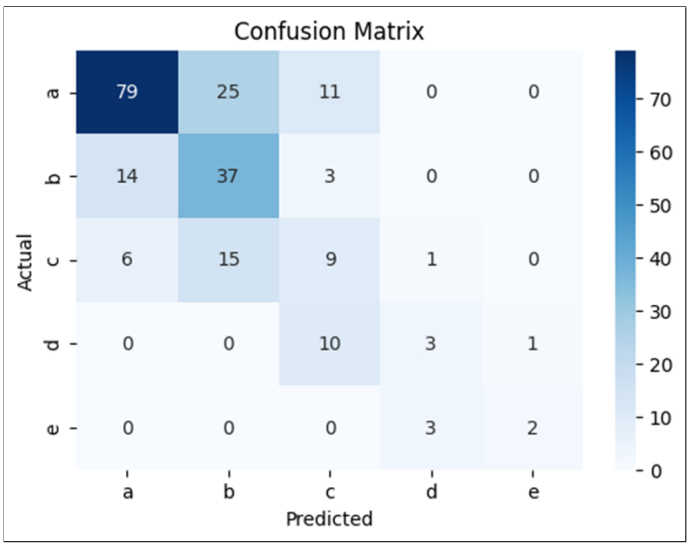

Figure 2.6: Additive Model: Confusion matrix with original label

confusion matrix reveals that our model predominantly predicted data in categories a and b akin to the original model, indicating alignment in classification within these classes. This observation suggests consistency in the classification patterns between our model and the original labels within these specific categories.

To enhance our comprehension of the outcomes and grasp the distinctions in grading between our model and the initial model, we allocated grades to diverse products and analyzed the outcomes. To illustrate, consider the following instance:

	Gerblé - Sugar Free Sesame Vanilla Cookie, 132g	Tartines de pain - blé complet - Bio - Pasquier_ - 240g	Amandes 100Cal la poignée - Carrefour - 200
Fat	Moderate	Moderate	High
Saturated fat	Moderate	Low	Moderate
Sugar	Low	Low	Low
Salt	Moderate	Moderate	Low
Energy	Moderate	Low	High
Original Model Grade	A	B	A
Our Model	B	A	C

Table 2.2: Nutritional Information for Various Products

Even though the third product (Amandes) exhibited higher levels of fat and saturated fat, the original NutriScore classified the product with an A grade. However, in our assessment, it received a C grade. This example from the product samples exemplifies why we have confidence in the reliability of our model. It showcases our model's enhanced capacity to manage correlations between energy and other constituents effectively.

Chapter 3

Electre-Tri Model

3.1 Building Profiles

In constructing profiles for the Electre-Tri model, percentile-based divisions serve as a fundamental approach and this section delineates the process of profile creation.

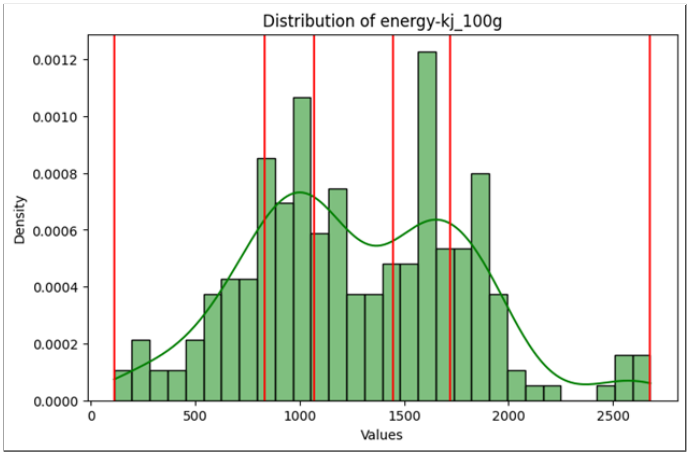

Figure 3.1: Electre-Tri Model: Energy profile

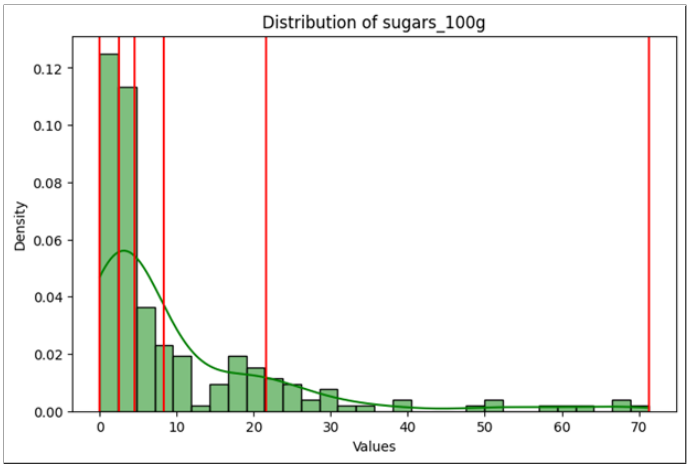

Figure 3.2: Electre-Tri Model: Sugar profile

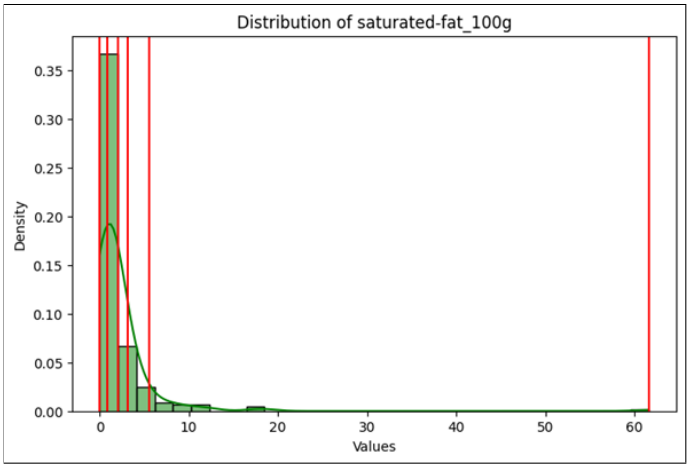

Figure 3.3: Electre-Tri Model: Fat profile

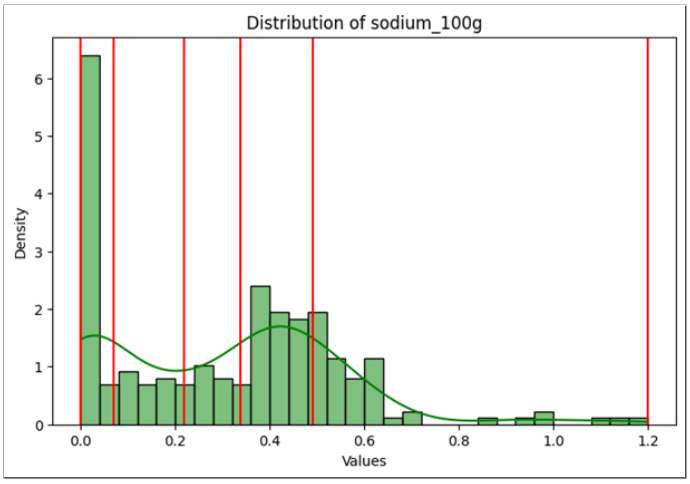

Figure 3.4: Electre-Tri Model: Salt profile

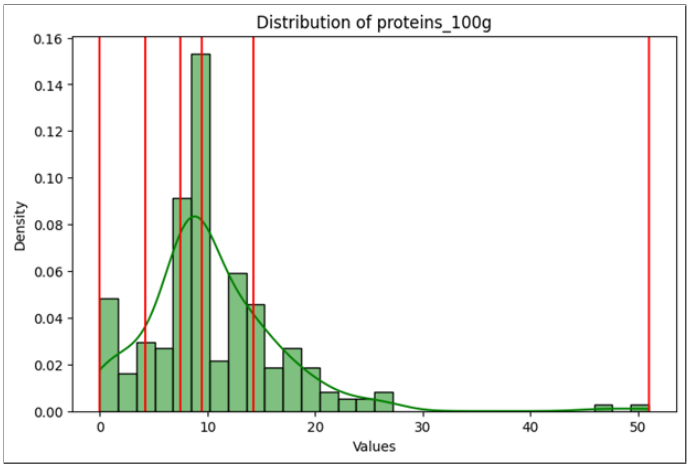

Figure 3.5: Electre-Tri Model: Protein profile

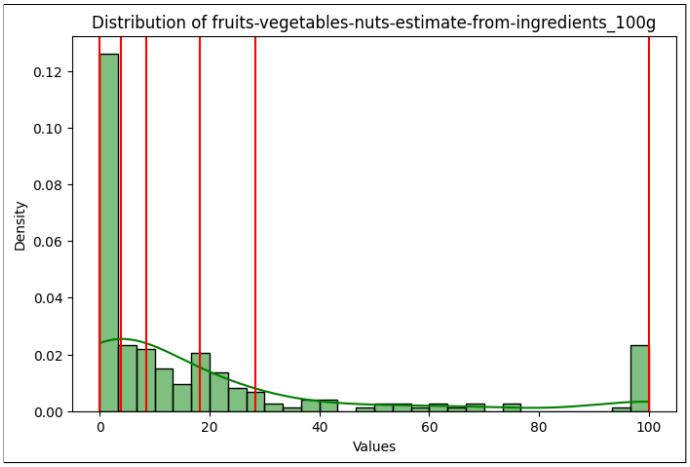

Figure 3.6: Electre-Tri Model: Fruits profile

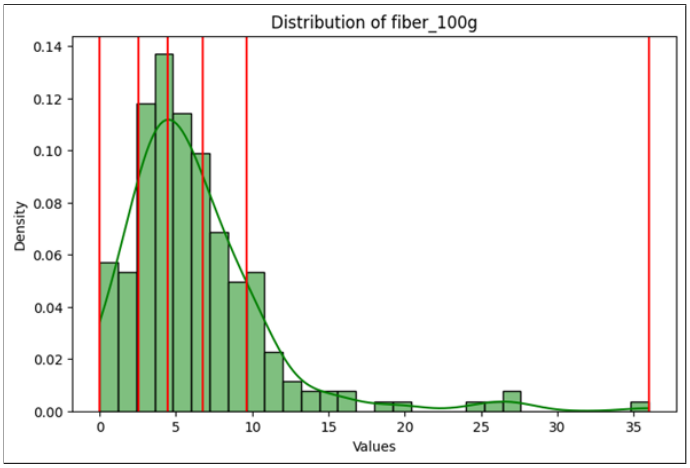

Figure 3.7: Electre-Tri Model: Fiber profile

Profiles are built by dividing the dataset into 20% intervals, ensuring a comprehensive yet manageable segmentation approach. The consolidated summary of 5 profiles (a, b, c, d, e) - thus giving 6 limits is represented in the Table 3.1. Furthermore, Figures 3.1 till 3.7 visually depict the distribution of nutrition values prevalent across the database, aiding in understanding the spread, central tendencies, and variability of these attributes, thereby proving crucial for defining profiles and establishing preference thresholds.

Certain criteria, such as saturated fat and proteins, exhibit right-skewed distributions. This skewness suggests that these criteria tend to aggregate more values toward their lower limits, resulting in a concentration of data points closer to the lower threshold. This contrasts with attributes following a normal distribution, indicating a clustering of profiles with similar lower limits for these skewed criteria.

	Energy	Sugar	Fat	Salt	Proteins	Fiber	Fruit
Profile Limit #1	111.0	0.0	0.0	0.0	0.0	0.0	0.0
Profile Limit #2	832.2	2.54	0.8	0.0704	4.2	2.546	3.90
Profile Limit #3	1070.4	4.58	2.0	0.2176	7.5	4.48	8.455
Profile Limit #4	1446.4	8.34	3.2	0.3376	9.5	6.7399	18.211
Profile Limit #5	1720.6	21.6	5.6	0.4904	14.3	9.620	28.286
Profile Limit #6	2679.0	71.3	61.6	1.2	51.0	36.0	100

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

3.2 Electre-Tri Models

3.2.1 Weights and Thresholds

This section talks about the two distinct sets of weights on which the models will be evaluated. The first set aligns with the weights provided in the problem statement (Table 3.2), while the second set corresponds to our tailored weights, resembling those utilized in the additive model (Table 3.3), but excluding energy (as the distribution of energy is a normal distribution as compared to other nutrients). This comparison will enable us to assess the impact of differing weight configurations on the performance and outcomes of the models.

Nutrition	Energy	Sugars	Fats	Salt	Proteins	Fiber	Fruits
Weight	1	1	1	1	2	2	2

Table 3.2: Electre-Tri Model: Provided weights

Nutrition	Energy	Sugars	Fats	Salt	Proteins	Fiber	Fruits
Weight	0	2	4	0.8	2	1	0.5

Table 3.3: Electre-Tri Model: Our weights

Also, in our evaluation, we will assess the models across three distinct thresholds: 50%, 60%, and 70%. These thresholds are derived by multiplying the threshold value (λ in percentage) by the product of the total number of profiles for the model and by the sum of weights, divided by 100, as represented in Equation 3.1.

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

3.2.2 Different Models

Within this project's scope, we will assess two Electre-Tri models: optimistic and pessimistic. These evaluations will employ various combinations of weights and thresholds as detailed in earlier sections. Consequently, our assessment will encompass twelve distinct evaluations (2 models $\times$ 2 weights $\times$ 3 thresholds).

3.3 Results and Analysis

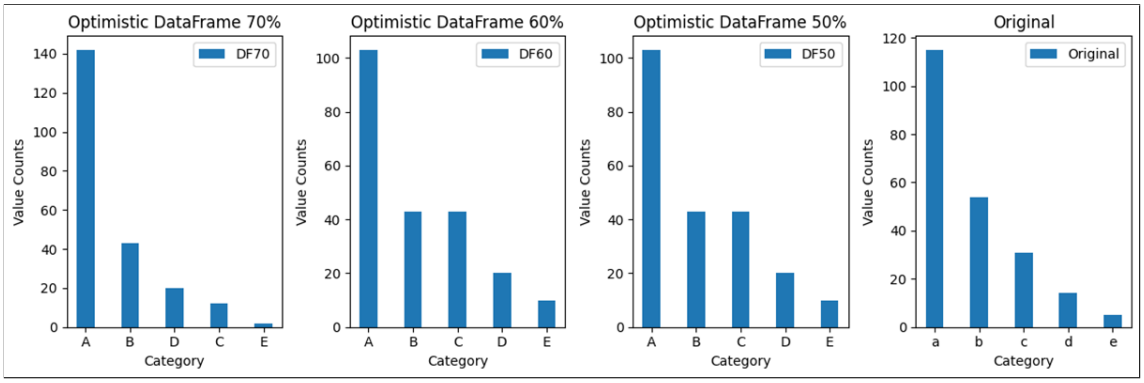

Figure 3.8: Electre-Tri Model: Optimistic with provided weights

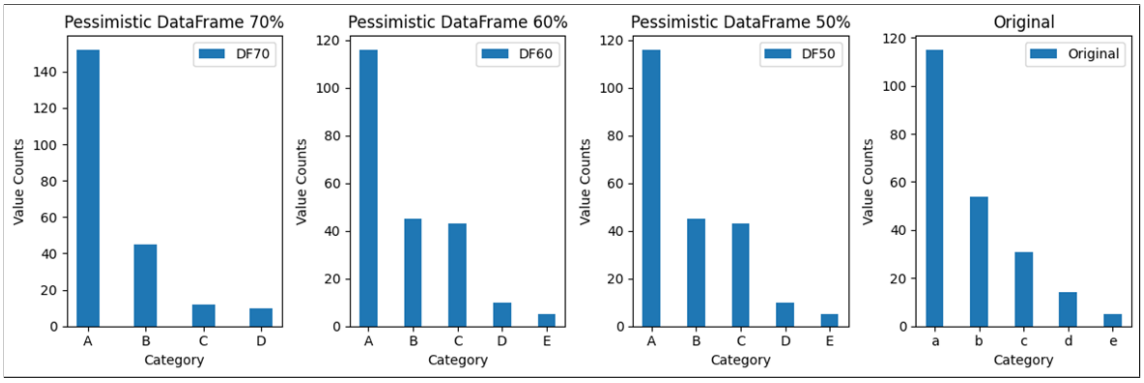

Figure 3.9: Electre-Tri Model: Pessimistic with provided weights

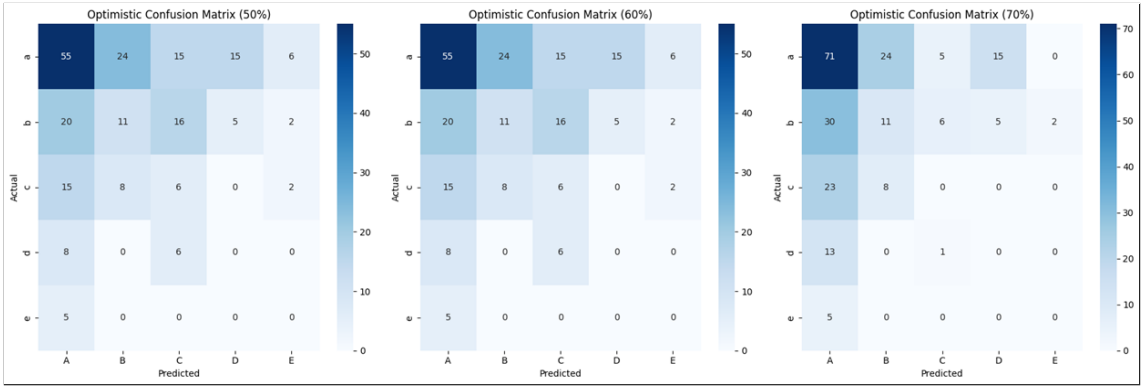

Figure 3.10: Electre-Tri Model: Confusion matrix optimistic with provided weights

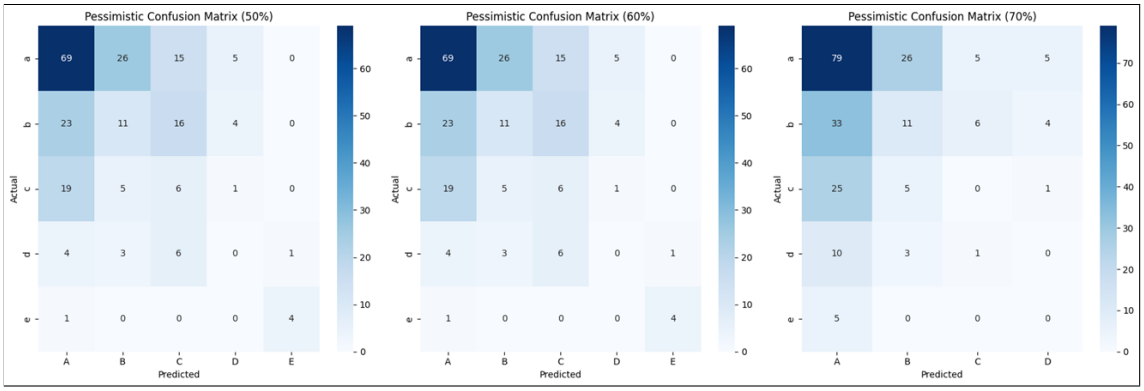

Figure 3.11: Electre-Tri Model: Confusion matrix pessimistic with provided weights

Figures 3.8 and 3.10 display the outcomes derived from assessments conducted using the optimistic model with provided weights as in Table 3.2. In contrast, Figures 3.9 and 3.11 showcase the findings obtained through evaluations employing the pessimistic model with provided weights. The subsequent insights encapsulate the analysis and interpretations drawn from the obtained results:

- **Optimistic model with provided weights**: The comparison between thresholds set at 50% and 60% revealed strikingly similar outcomes in the model's classifications. This similarity, despite the variation in thresholds, suggests a potential convergence or plateauing effect in the model's decision-making at these specific threshold levels.

  However, an intriguing observation emerged when considering the impact of skewed data on the system's behavior. The presence of skewed data exerted a significant influence, manifesting in the system's inclination toward an overwhelmingly optimistic grading approach. This bias led to a consistent tendency of the model to assign grades predominantly toward the higher classes, particularly a and b.

- **Pessimistic model with provided weights**: A key observation is the similarity between the outcomes generated at threshold levels of 50% and 60%. These results exhibit striking resemblance, suggesting a convergence or overlap in the classifications made by the model at these particular threshold points. Such consistency across these thresholds indicates a strong alignment in the criteria utilized for the classification of the dataset.

  An intriguing anomaly surfaced when employing the Pessimistic model with a 70% threshold. Surprisingly, this configuration failed to assign any item to grade E. Such a result raises questions about the model's behavior and decision-making process. This outcome signifies a stringent set of criteria or limitations imposed by the model's configuration, resulting in an absence of items categorized under grade e.

  Notably, the Pessimistic model, especially when employed with higher thresholds, exhibited a distinct skewing of grades toward the topmost class, i.e., class a. This disproportionate skewing signifies an inherent inclination of the

model to assign more items to the higher-grade categories. The rise in stringency caused by the higher threshold settings might have influenced this trend, potentially leading to an overemphasis on stringent criteria and thus a predominance of items classified within class a.

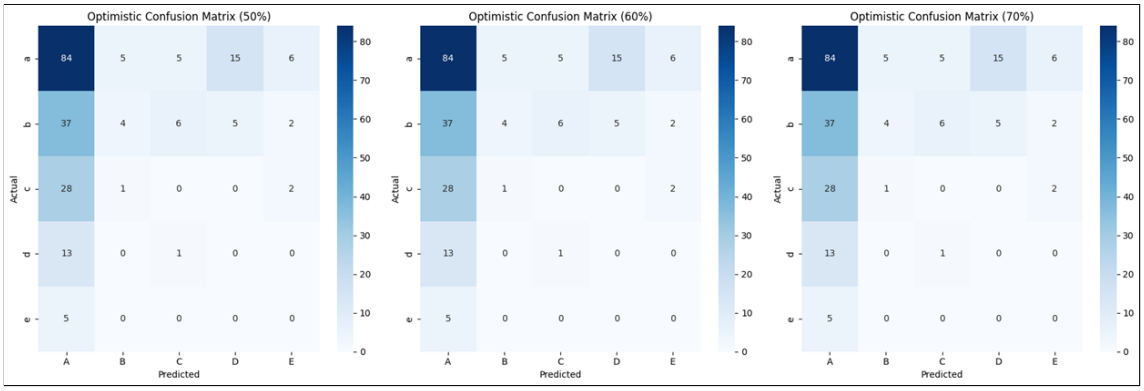

Figure 3.12: Electre-Tri Model: Confusion matrix optimistic with our weights

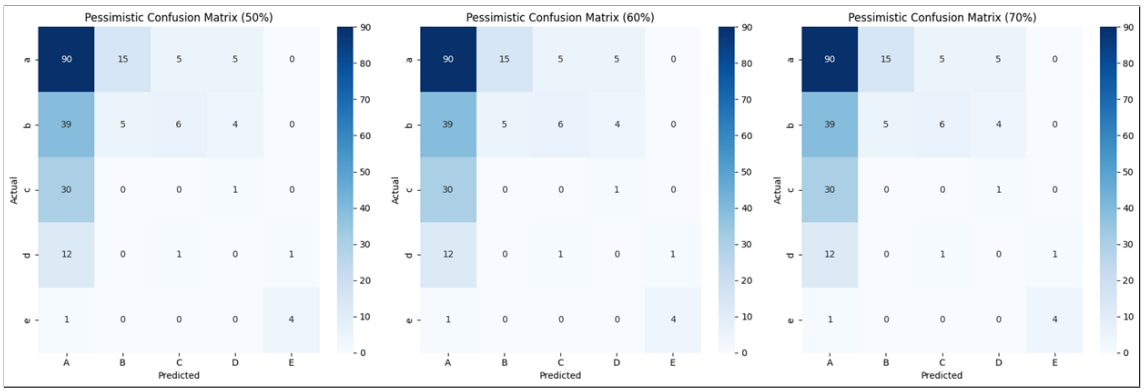

Figure 3.13: Electre-Tri Model: Confusion matrix pessimistic with our weights

Figures 3.12 and 3.13 illustrate the outcomes obtained from the application of confusion matrices for both the optimistic and pessimistic models executed based on our chosen weights as in Table 3.3. Some insights gained through these visuals are as follows:

- An intriguing trend emerges irrespective of the designated threshold value: the classification of items into their respective classes remains consistent. This emphasizes robustness in the classification process that exceeds the threshold selection.
- Despite the inherent ideological differences in approaches of optimistic and pessimistic models, the difference in outcomes between these models appears notably marginal. This revelation suggests potential convergence or similarity in their assessments.
- The convergence in outcomes between these models is quite striking, especially when considering a vast majority—approximately 95%—of the products, underscoring their reliability and consistency in providing classifications.

Chapter 4

Machine Learning Models

4.1 Decision Tree

In configuring the Decision Tree, the criterion selected for splitting nodes was **entropy**, aiming to determine the most informative features at each stage. Additionally, the maximum depth of the tree was set at **7**, limiting the tree's growth to prevent overfitting and maintain interpretability. Figure 4.1 showcases the confu-

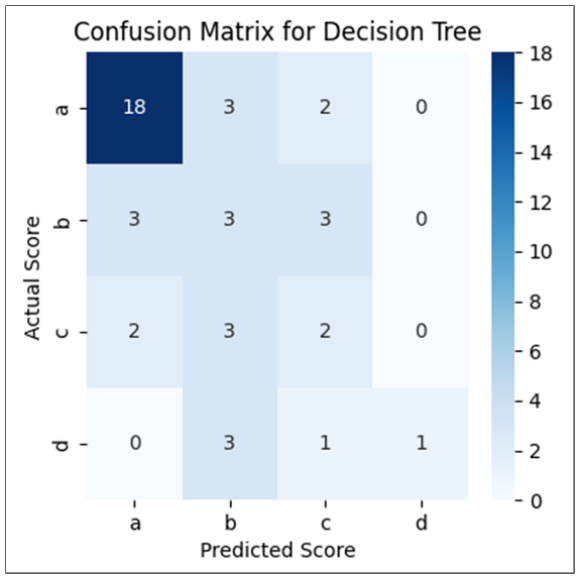

Figure 4.1: Machine Learning Model: Decision Tree

sion matrix resulting from the Decision Tree's evaluation on our dataset, providing a visual representation of the model's performance, detailing the counts of correct and incorrect classifications across different classes.

The resulting analysis reveals a noticeable bias in the classification outcomes, significantly influenced by the prevalence of class occurrences within the dataset. Specifically, the data is skewed, showcasing a substantial presence of products falling into classes a, followed by b, c, and then d. However, it's essential to note a limitation observed in the model's predictive capability. It faced challenges in accurately predicting items belonging to class e, possibly due to the scarcity or lack of distinct features defining this particular class.

The accuracy achieved by the Decision Tree model stands at 54.5%. This quantifies the proportion of correctly predicted classifications given the total number of items evaluated. While the model demonstrates some predictive capability, the accuracy rate indicates a moderate performance level, highlighting room for improvement in accurately classifying items based on their features.

4.2 Random Forest

The Random Forest model was configured with a maximum depth limited to **5** restricting overfitting of the tree's growth. Figure 4.2 depicts the confusion matrix, illustrating the outcomes of the Random Forest model's predictions on our dataset. In comparison to the Decision Tree model, the Random Forest exhibited reduced

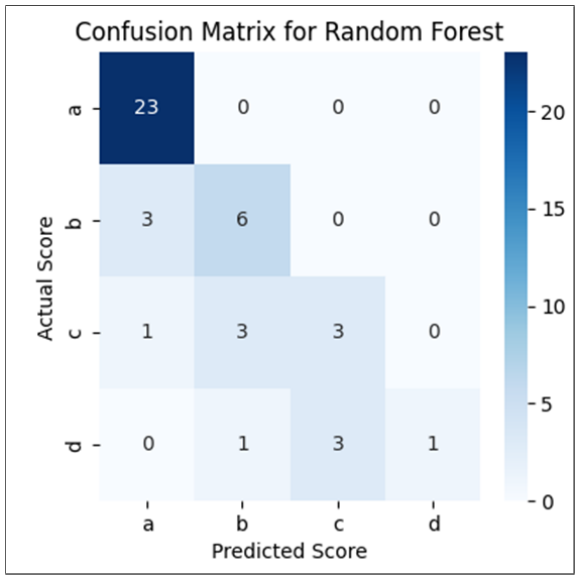

Figure 4.2: Machine Learning Model: Random Forest

bias against the dataset, signifying a better adaptation to the dataset's intricacies and complexities, allowing for more flexible and accurate predictions.

An intriguing observation from the model's predictions lies in the nature of class assignments to products. The model tends to assign a product either to the same class or a lower class but seldom predicts a higher class. This behavior suggests a cautiousness in classifying products, ensuring a tendency to avoid overestimation. However, the model encounters limitations in predicting instances belonging to class e. This inability to predict class e points to a specific challenge or complexity within the dataset that the model struggles to discern accurately.

Regarding the model's overall accuracy, it achieves a commendable accuracy rate of 75%. This accuracy score indicates the model's ability to correctly classify products into their respective classes in three out of every four instances, showcasing its efficiency in handling the dataset.

4.3 Logistic Regression

The decision to apply **default** configurations to the Logistic Regression model was based on their broad applicability across diverse datasets. These settings offer simplicity, aiding straightforward evaluation and comprehension of the model's behavior. Simultaneously, default configurations help mitigate the risk of overfitting, ensuring a balance between model complexity and generalization. The outcomes of the Logistic Regression model's predictions on our dataset are represented in Figure 4.3, which showcases the confusion matrix. Similar to the Decision Tree model, the

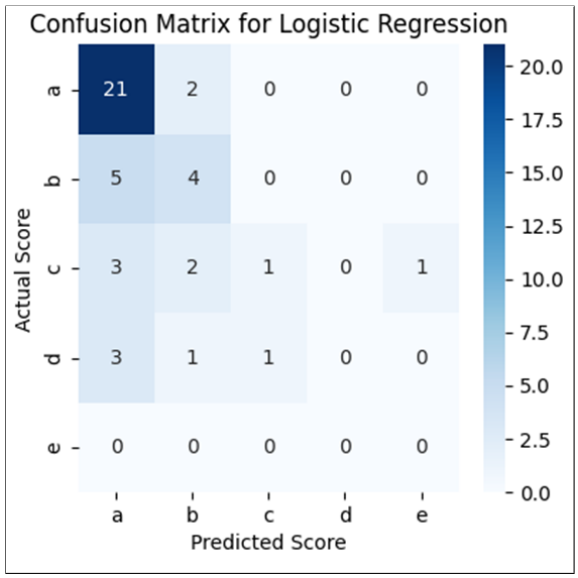

Figure 4.3: Machine Learning Model: Logistic Regression

Logistic Regression model exhibits several parallels in its behavior and outcomes. This similarity suggests resemblances in their predictive capabilities. However, a distinctive observation emerges from the Logistic Regression model's predictions that it showcases a bias towards assigning products to the a class. This inclination could be attributed to the dataset's composition, where a significant portion comprises products belonging to the a class which is why, the model, tends to favor assigning products to the a class more frequently.

While the Logistic Regression model does predict instances belonging to class e, these predictions are often incorrect highlighting a limitation in accurately classifying products into class e. This also suggests the complexities within class e that the model struggles to capture effectively.

Despite these observations, the model achieves an overall accuracy rate of 59.1%. This accuracy score highlights the model's ability to make correct predictions in approximately 59 out of every 100 instances, showcasing its moderate efficiency in handling the dataset.

4.4 Gaussian Naïve Bayes

The Gaussian Naïve Bayes model was utilized with **default** settings to adhere to simplicity and initial assessment given the small size of the dataset. The confusion matrix, depicting the results of the Gaussian Naïve Bayes model's predictions on our dataset, is illustrated in Figure 4.4. As opposed to the Decision Tree comparison,

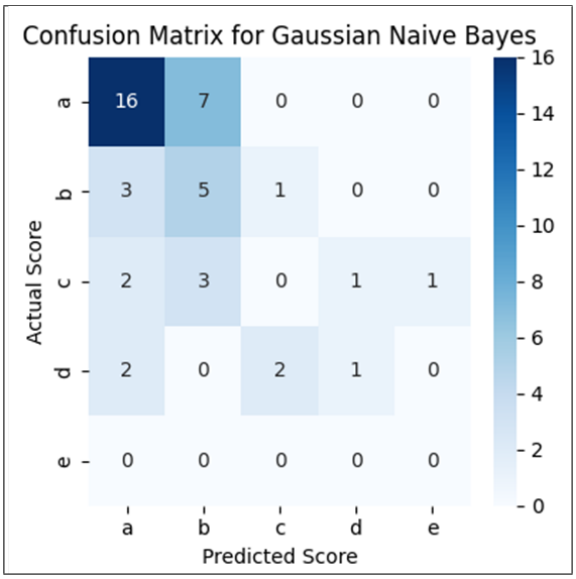

Figure 4.4: Machine Learning Model: Gaussian Naïve Bayes

the Gaussian Naïve Bayes model showcased reduced bias against the dataset, indicating a more adaptable approach to the dataset's nuances, resulting in predictions that align better with the data distribution.

An important trend observed in the model's predictions is its tendency to assign a product to either the same class or a lower class, rarely predicting a higher class. This cautious approach suggests a tendency to avoid overestimation, ensuring conservative predictions. However, despite this cautiousness, the model displays an error in predicting instances belonging to class e. This misclassification indicates a specific challenge within the dataset that the model struggles to accurately discern when assigning class labels.

In terms of overall accuracy, the Gaussian Naïve Bayes model achieves a commendable accuracy rate of 75%. This accuracy signifies the model's ability to correctly predict the class labels for three out of every four instances, demonstrating its effectiveness in handling the dataset.

4.5 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) model was configured with the number of neighbors set to **15**. A higher value of K (in our case, since it is 15), typically smoothens the decision boundary, leading to a more generalized model but potentially blurring distinctions between classes, causing the model to lean more towards the prevalent classes in the dataset as we will see in the further explanation.

The confusion matrix illustrating the results of the K-Nearest Neighbor model's pre-

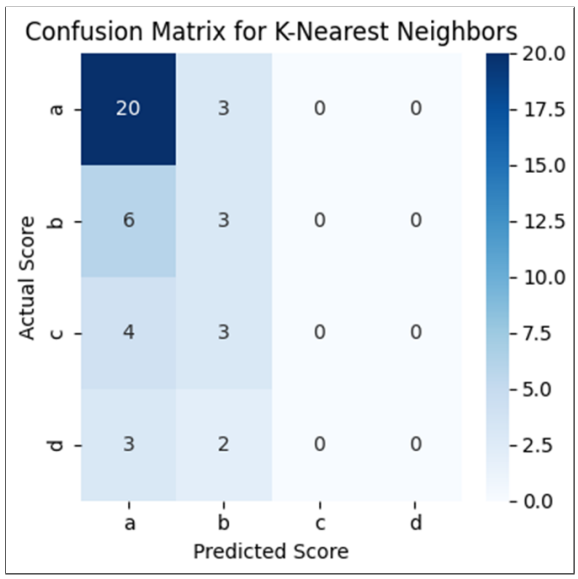

Figure 4.5: Machine Learning Model: K-Nearest Neighbors

dictions on our dataset can be observed in Figure 4.5. Comparing its performance to the Decision Tree model, KNN displays similarities, suggesting resemblances in their predictive behavior. One notable observation from the KNN model's predictions is its inclination to bias results toward the a class, reflecting the dataset's composition, as it contains a substantial number of products categorized under class a. Thereby, the model tends to favor this prevalent class in its predictions.

However, due to the specific setup where the number of neighbors is greater than the occurrences of class e within the dataset, the model tends to treat instances of class e as outliers, suggesting a limitation in the model's ability to correctly predict or generalize this particular class.

But, the overall accuracy of the KNN model stands at 52.3%. This accuracy rate indicates the model's ability to correctly classify products in slightly more than half of the instances, showcasing a moderate level of predictive capability.

4.6 Comparative Results and Analysis

For evaluating different machine learning models using the same database, we split the data into **train and test sets** with the 80% and 20% ratio respectively. Thus, the train set turned out to be 176 data points while the test set had 44 data points. Figure 4.6 depicts the accuracy of various classifiers throughout the dataset. The performance of machine learning models is heavily reliant on **data quantity and quality**. A crucial aspect is the necessity for **unbiased representation** across the various classes within the dataset. The models learn from the existing data, and any biases or correlations present in the original dataset tend to persist within the models' predictions. Among the models evaluated, the **Random Forest** model stands out with the **highest accuracy** compared to other models examined in this

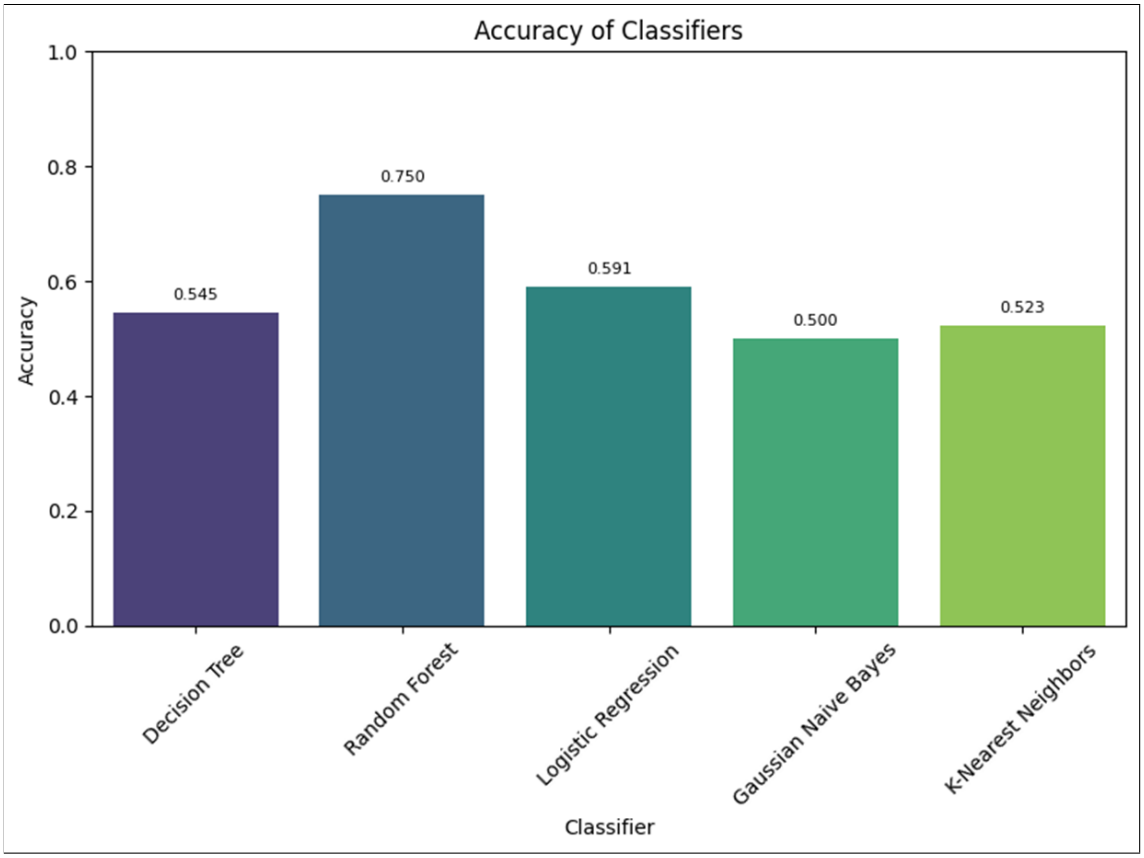

Figure 4.6: Machine Learning Model: Accuracy comparison of all models

analysis, highlighting the effectiveness of this model in making precise predictions within the dataset.

Similar to earlier observations, a **consistent trend** is observed across models where the assigned classes to products tend to remain either the same or lower but rarely predict a higher class. This conservative approach ensures a cautious prediction strategy to **avoid overestimation** or misclassification.

A critical insight gleaned from the analysis is the **positive impact of increased data volume**. More data tends to enhance true positive predictions while reducing false negatives, thereby refining the models' accuracy and reliability in accurately categorizing products into their respective classes.

Chapter 5

Comparison with another group

In this chapter, we present a comparative analysis between our results and those obtained by another group. The distribution of labels in the alternative group is depicted in Figure 5.1.

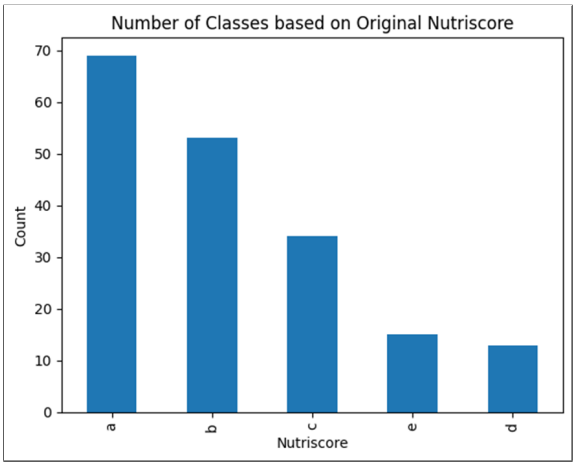

Figure 5.1: Comparison: Original class distribution

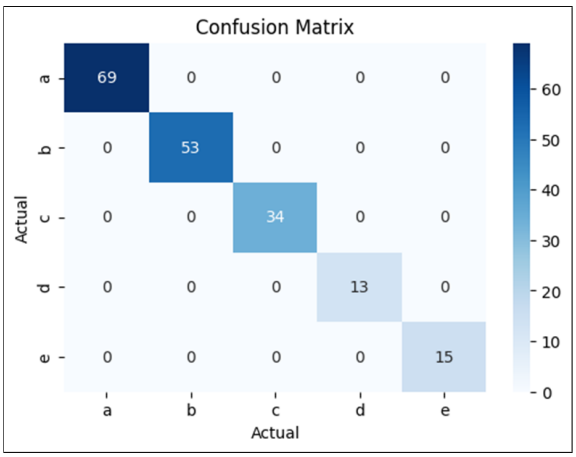

Figure 5.2: Comparison: Confusion matrix of original class distribution

Additionally, Figure 5.2 illustrates the confusion matrix corresponding to their original class distribution. Notably, the alternative group exhibits a higher number

of classes, particularly in the class e, in contrast to our dataset. Furthermore, we observed the inclusion of beverages in their dataset, an element absent in ours, complicating the generalization of our model's performance.

On applying our additive model as in Table 2.1 with our weights, we see the results summarized in Figure 5.3. Notably, the results achieved had an accuracy of

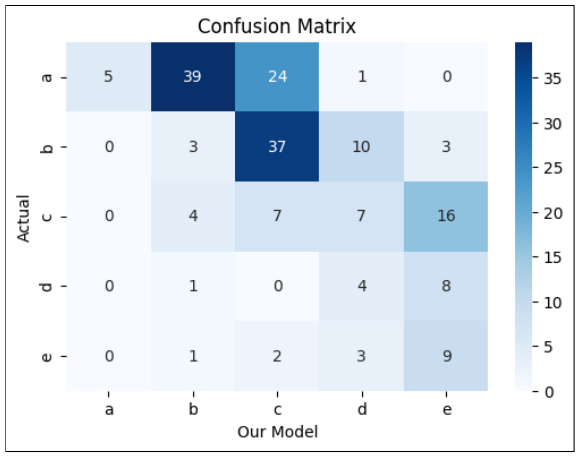

Figure 5.3: Comparison: Our additive model vs original label

true positives (TP) as follows:

- Class a: 7.2%
- Class b: 1.8%
- Class c: 20.5%
- Class d: 30.8%
- Class e: 60%

This resulted in an overall accuracy of 24.06%. To further contextualize our findings, we compare our results with another group's additive model applied to their dataset, as displayed in Figure 5.4. Notably, the results achieved had an accuracy of true positives (TP) as follows:

- Class a: 0.0%
- Class b: 37.74%
- Class c: 58.9%
- Class d: 15.4%
- Class e: 0.0%

The overall accuracy for their model stands at 22.41%.

Before presenting our final comparative results, we visualise our additive model's results and their additive model's results on their dataset in Figure 5.5. Through

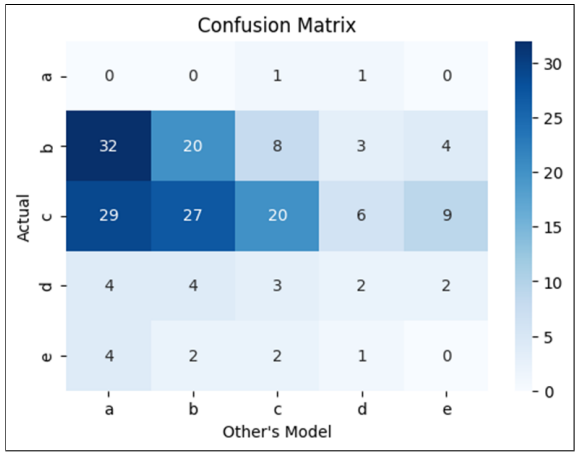

Figure 5.4: Comparison: Their additive model vs original label

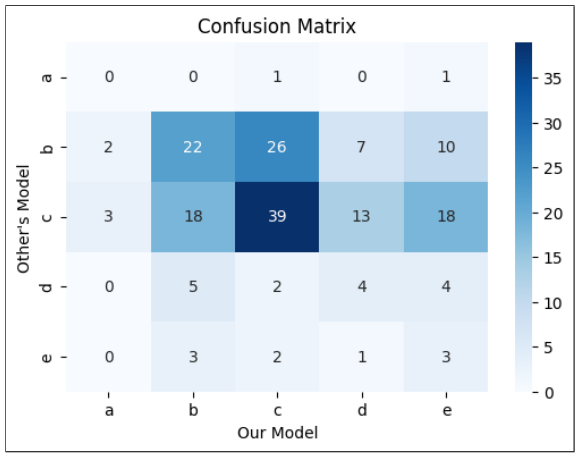

Figure 5.5: Comparison: Our additive model vs their additive model

meticulous analysis, it's evident that our model excels in providing balanced accuracy across all classes without neglecting any specific category. However, it is essential to note that the other group's model showcases superior performance, particularly in the intermediate classes (B & C). This comparative study underscores the significance of a nuanced approach to model evaluation. While our model demonstrates overall consistency, the inclusion of beverages in the alternate dataset poses challenges for generalizability. Therefore, considering the exclusion of this category could potentially refine the evaluation process, enabling a more accurate assessment of our model's performance.

Chapter 6

Conclusion

In this project focused on decision modeling for the Nutri-Score label, we embarked on a comprehensive exploration of various methods to evaluate and assign nutritional scores to food products. The Nutri-Score, a system designed to simplify nutritional information for consumers, served as the basis for our analysis. We approached this task from multiple angles, considering the Nutri-Score as a Multi-Criteria Decision Analysis (MCDA) problem and delving into different models to assess its computation and application. The project unfolded in several stages, each aimed at developing and refining models for evaluating the Nutri-Score.

- **Constructing Databases:** We developed a subset of data from the huge database of foods by leveraging available resources for nutritional information.
- **Additive Models:** We constructed additive models to compute Nutri-Scores, carefully selecting weights and justifying their choices. We also compared these scores to the original Nutri-Score for validation.
- **Electre-Tri Models:** We explored PessimisticmajoritySorting and OptimisticmajoritySorting based on predefined weights and thresholds. These models categorized foods into Nutri-Score labels and were tested against the real Nutri-Score assignments for analysis.
- **Machine Learning Approach:** Leveraging machine learning algorithms, we developed AnotherMethodNutriScore to assign Nutri-Score classes to foods. We compared these results with previous models and the real Nutri-Score assignments.
- **Comparison and Evaluation:** Additionally, we extended our analysis to compare our MCDA model (additive model specifically) with that of another group, aiming for a comparative assessment of their respective performances.

This multifaceted project encompassed diverse methodologies, from MCDA principles to machine learning, catering to different aspects of Nutri-Score computation and classification.

We present a comprehensive exploration of decision models applied to the dataset and the models developed here offer valuable insights into the complexity of nutritional scoring systems and their applications in food labeling, catering to the diverse needs of consumers in making informed dietary choices.

Moreover, the visualizations and comparative analyses presented in this report highlight the importance of contextualizing model outcomes within specific dataset characteristics. This contextual understanding serves as a crucial factor in refining and enhancing the precision of decision models for real-world applications.

Ultimately, the comparative study performed not only contributes to understanding the strengths and limitations of different decision models but also emphasizes the need for tailored approaches in model evaluation, paving the way for more robust and accurate predictive frameworks in diverse domains.

The entire project can be found on the GitHub link here [5].

Bibliography

[1] https://world.openfoodfacts.org/, [Accessed 17-12-2023].

[2] *Choose more wisely with the nutri-score — colruyt group — colruytgroup.com*, https://www.colruytgroup.com/en/conscious-consuming/nutri-score, [Accessed 17-12-2023].

[3] *C'est quoi le nutri-score? — lejdd.fr*, https://www.lejdd.fr/Societe/cest-quoi-le-nutri-score-4072231, [Accessed 17-12-2023].

[4] J. Brabants, *Nutri-Score - A Simple Science-Based Nutritional Value Labelling System for the Food Industry — get.apicbase.com*, https://get.apicbase.com/nutri-score-science-based-nutritional-value-labelling-system/, [Accessed 17-12-2023].

[5] https://github.com/risg99/Nutriscore-Label-Decision-Modeling, [Accessed 22-12-2023].