DECISION MODELS FOR THE NUTRI-SCORE LABEL OF FOODS

Supervisor:
Prof. Brice MAYAG

Presented by:
Xianyun Zhuang
Jintao Ma
Yutao Chen

CONTENTS

01

Dataset

- Extraction
- Preprocessing
- EDA exploration

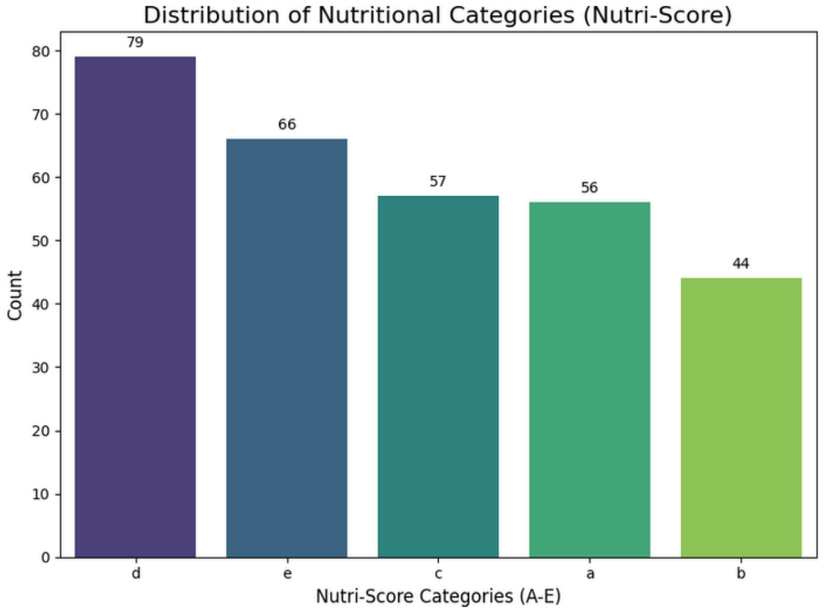
Dataset

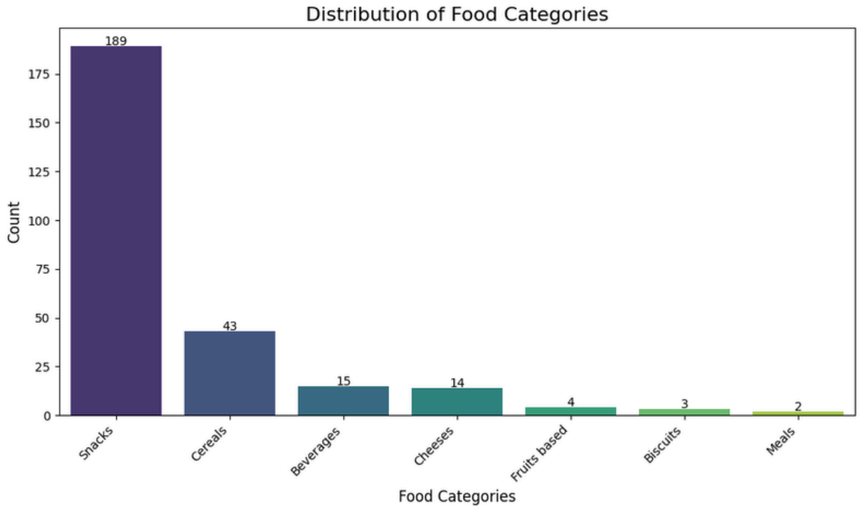
Dataset Overview:

- **Data Source**: Open Food Facts API
- **Country**： France
- **Categories Extracted**: 'Snacks', 'Biscuits', 'Cereals', 'Meals', 'Beverages', 'Cheeses', 'Fruits based'
- Ensure our database overlaps with other groups' databases by no more than 30%.

```
# check duplication rate
df = pd.read_excel("/kaggle/input/fffffffffffood/products.xlsx")
df_cleaned = pd.read_excel("/kaggle/input/fffffffffffood/food_data_cleaned1.

duplicates = df.duplicated(subset=['barcode'], keep=False)

duplicate_percentage = duplicates.sum() / len(df) * 100

if duplicate_percentage > 30:
    print(f"Warning: Duplicate rate is {duplicate_percentage:.2f}%")
else:
    print("Duplicate rate is acceptable.")
```

```
Duplicate rate is acceptable.
```





Data Preprocessing:

- Preprocessing Steps:
  - **Drop Missing Values**: Dropped rows with missing values
  - **Drop Duplicates**: Dropped duplicated rows
  - **Lable selection**:Selected one lable from multiple labes
  - **Final Dataset**: Filtered relevant columns for analysis, created a final preprocessed dataset with 300 randomly sampled items.

Comparison of two datasets

A function comparing two databases based on **barcode**:

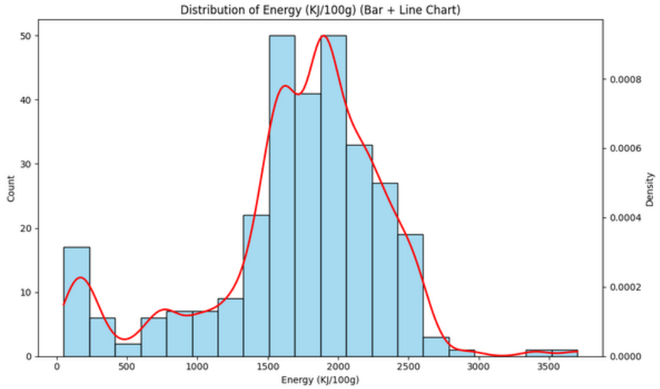
```
# Ensure 'Barcode' in file1 and 'barcode' in file2 exist
if 'Barcode' in data1.columns and 'barcode' in data2.columns:
    # Extract barcodes
    barcodes1 = set(data1['Barcode'].dropna().unique())
    barcodes2 = set(data2['barcode'].dropna().unique())

    # Calculate overlap
    overlap = barcodes1.intersection(barcodes2)
    overlap_percentage = (len(overlap) / min(len(barcodes1), len(barcodes2))) * 100
```

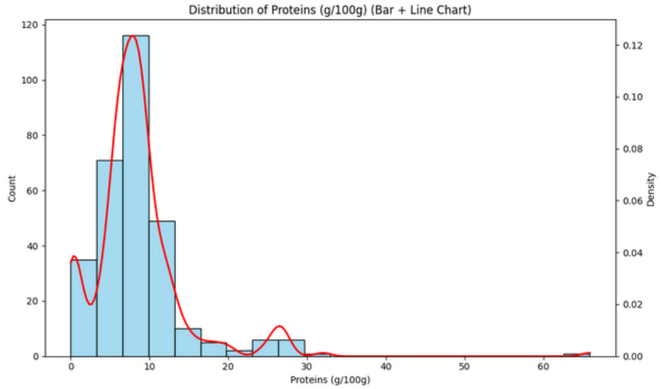
Reults:

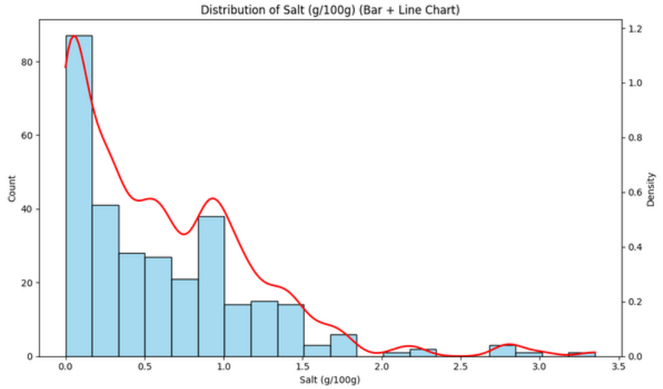
Total barcodes in File 1: 302
Total barcodes in File 2: 414
Number of overlapping barcodes: **38**
Overlap percentage: **12.58%**
Satisfy less than 30% requirement

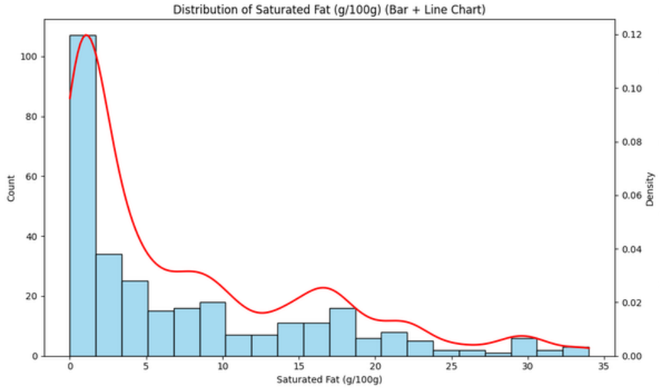
Dataset

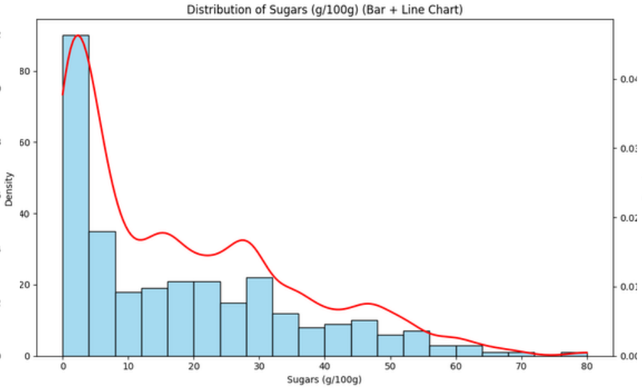
EDA exploration

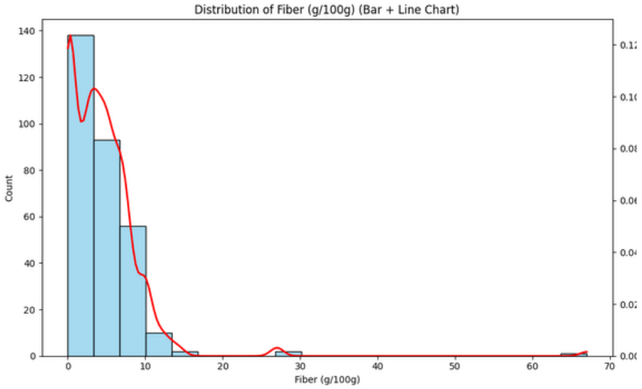


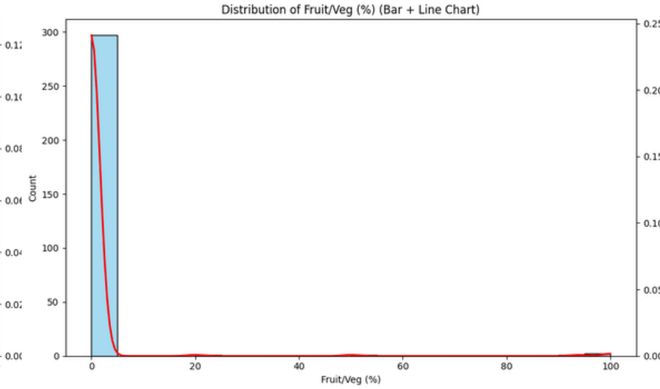










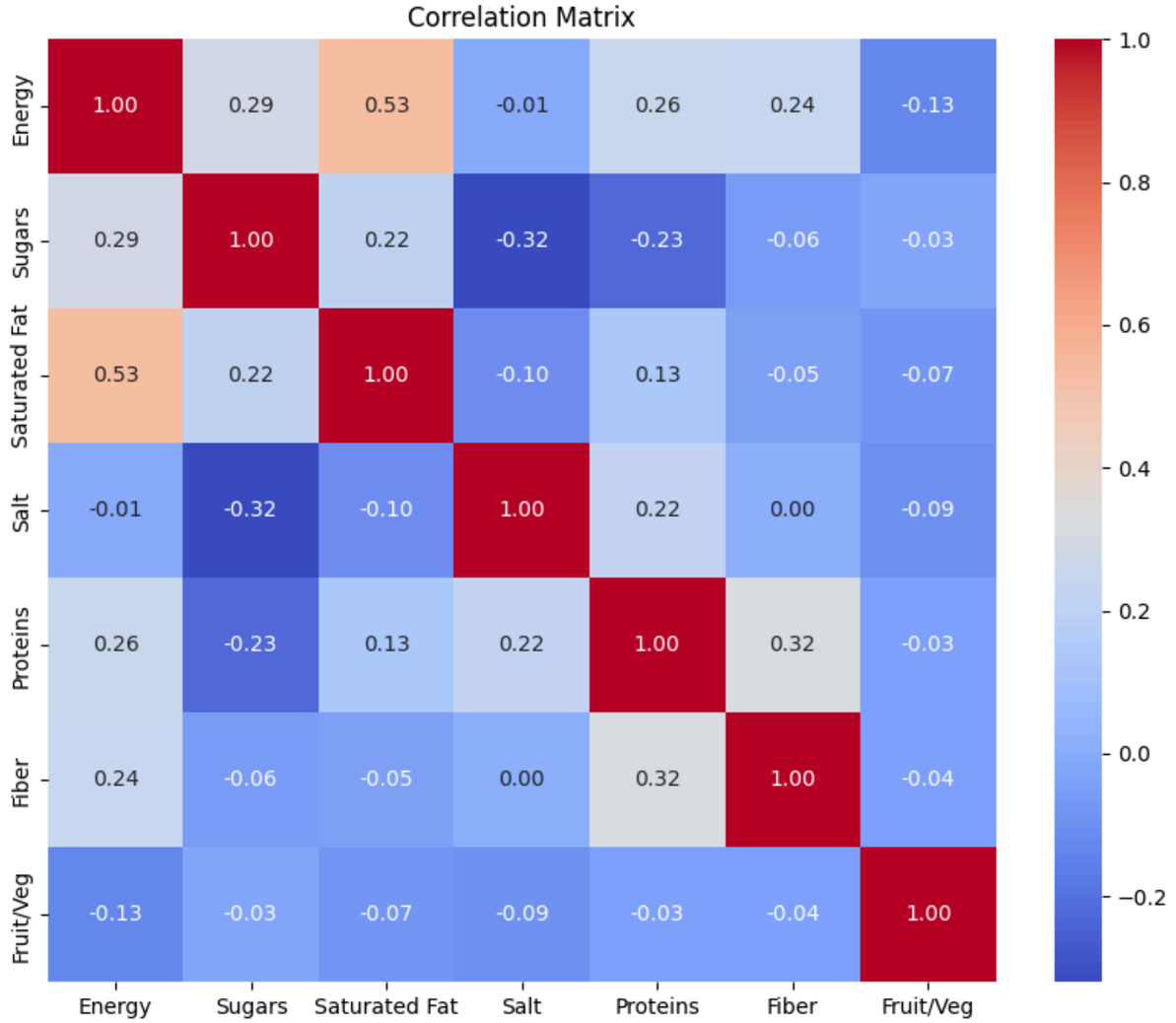


Dataset EDA exploration

EDA exploration:

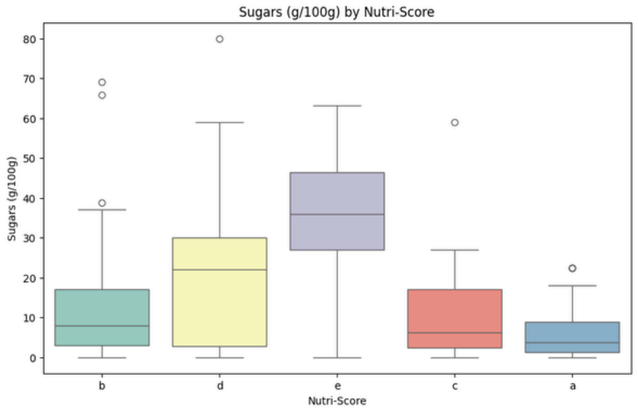
Inferences from the Matrix
Based on this matrix, we can draw the following conclusions:

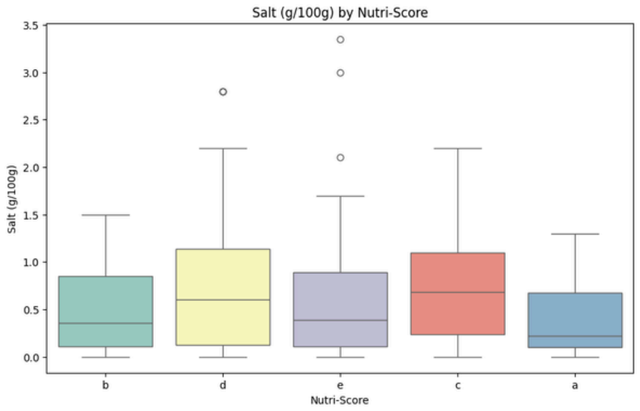
- Energy is positively correlated with sugars and saturated fat: This suggests that foods high in energy tend to be higher in sugars and saturated fats.
- Energy is negatively correlated with fruits/vegetables: This implies that high-energy foods generally have lower fruit/vegetable content.
- Sugars and saturated fats are positively correlated: This indicates that foods high in sugar tend to be higher in saturated fat as well.

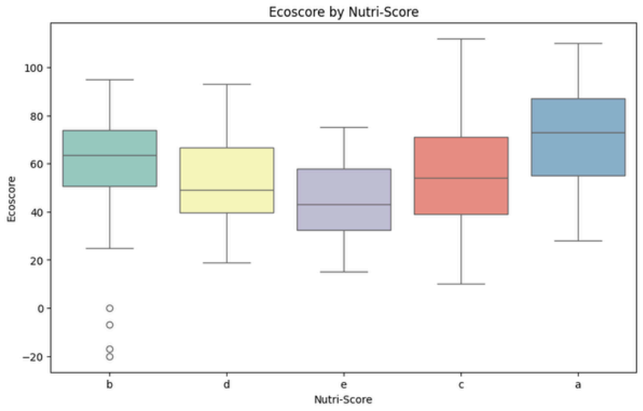


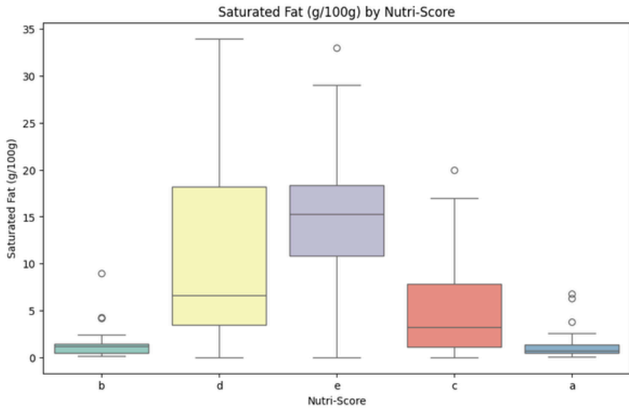
Dataset

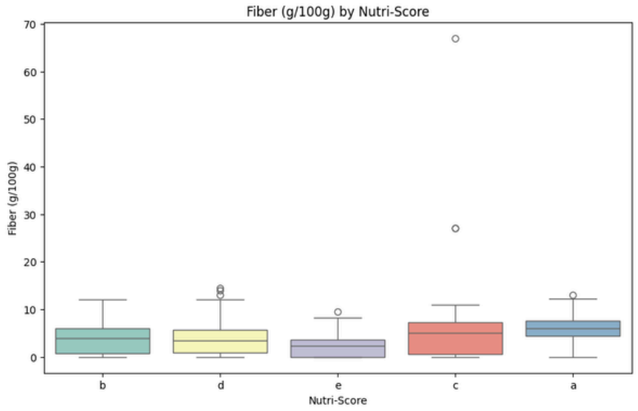
EDA exploration

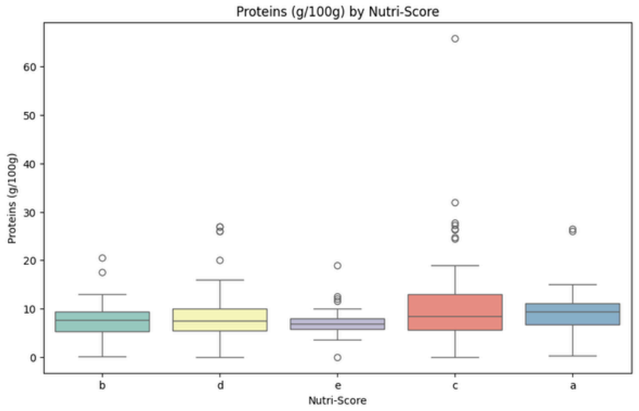


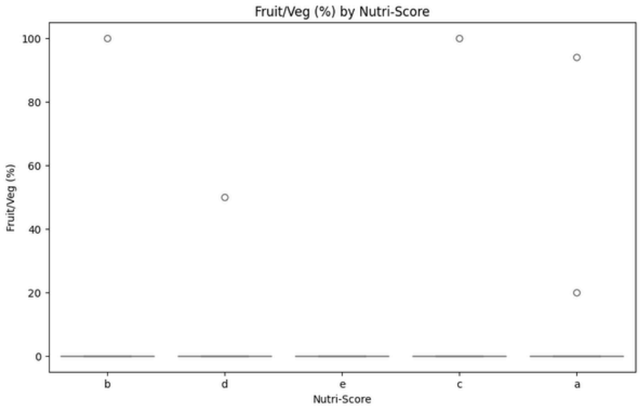


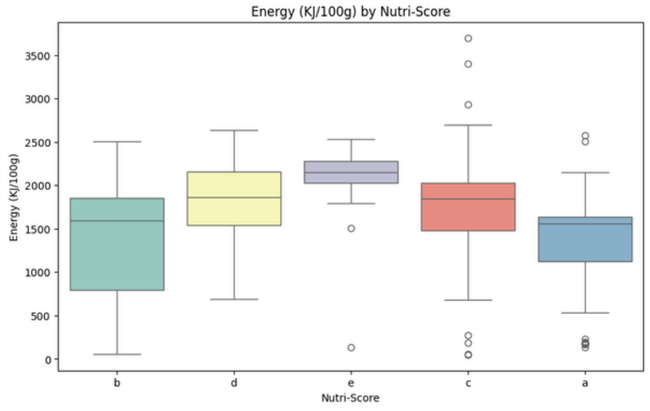












Dataset

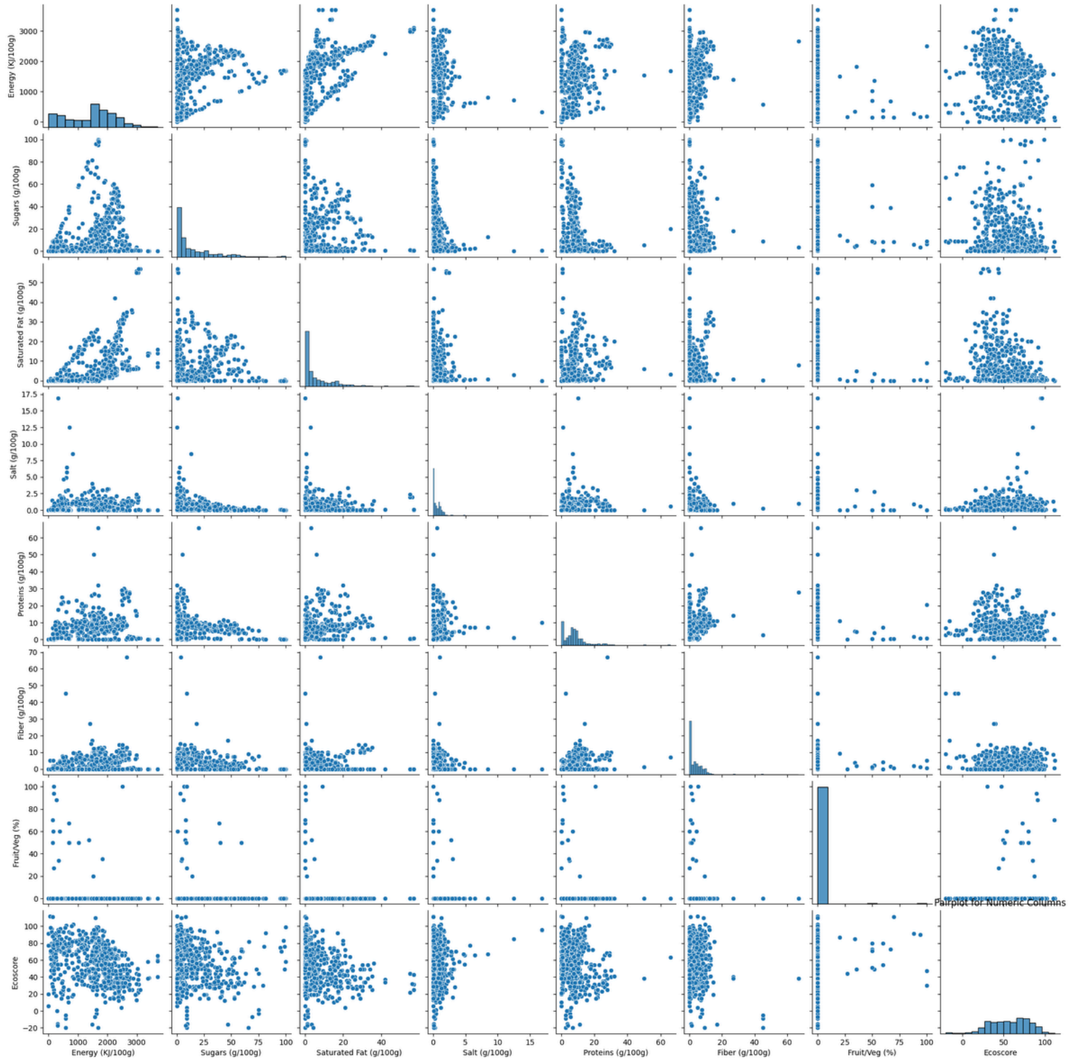
EDA exploration:

Positive Correlation Trend

- There might be a positive correlation between certain variables: for example, the scatter plot between Calories and Sugar may show an upward trend.
- Foods with higher sugar content often have higher calorie levels.

Clustering Phenomenon

- Some scatter plots between variables exhibit noticeable clusters of points: this may indicate that specific food types or categories are distinctly grouped within these variable ranges.
- e.g., beverages vs. snacks



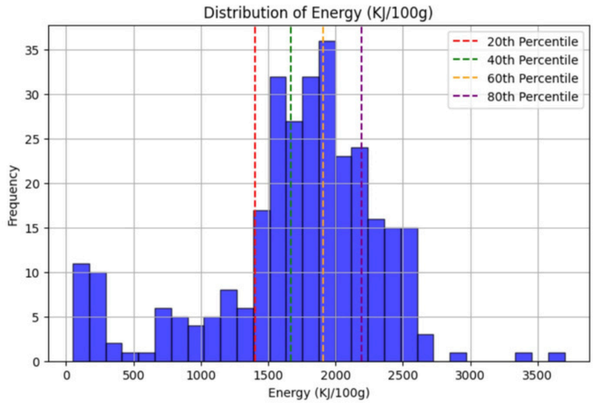
02

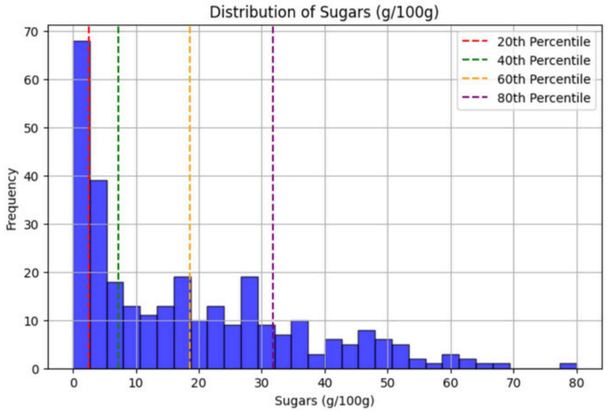
ELECTRE-Tri Model

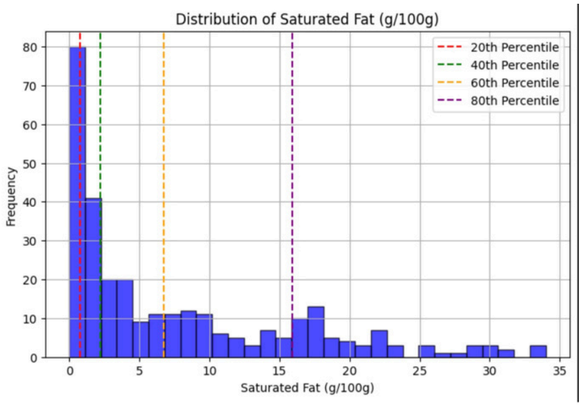
- Pessimistic
- Optimistic

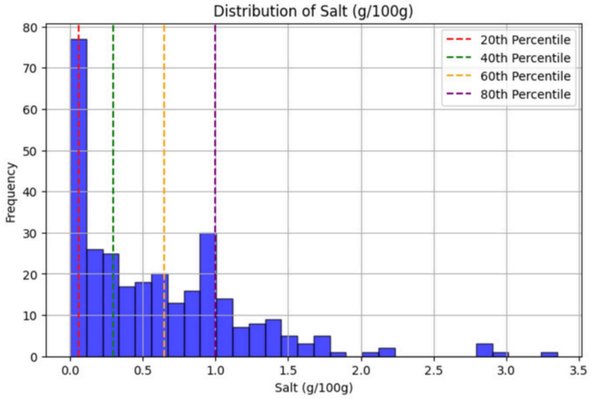
limiting profiles and data segmentation(quintiles)

Data Distribution(healthy and unhealthy)



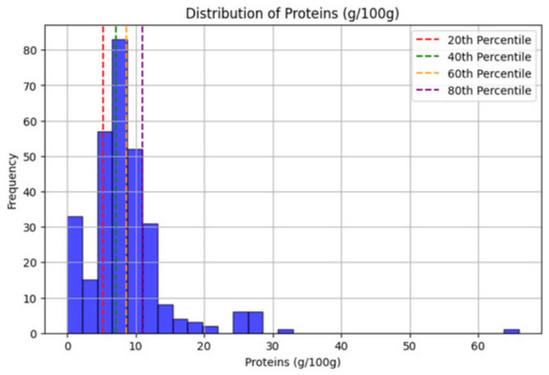


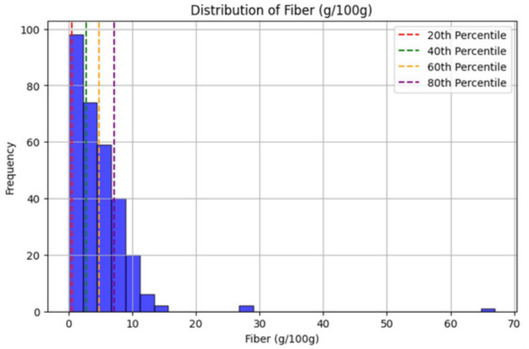


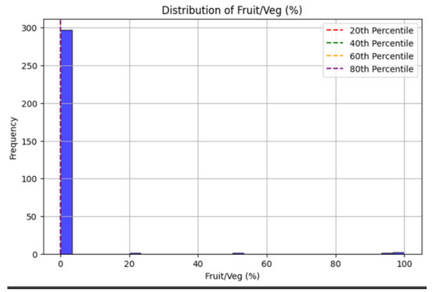


Minimization criteria	Energy	Sugars	Saturated Fat	Salt \
1	1402.6	2.50	0.80	0.06
2	1670.8	7.18	2.24	0.30
3	1905.6	18.60	6.76	0.65
4	2190.0	31.80	15.90	1.00

Maximization criteria	Proteins	Fiber	Fruit/Veg
1	5.3	0.50	0.0
2	7.1	2.80	0.0
3	8.6	4.80	0.0
4	11.0	7.13	0.0



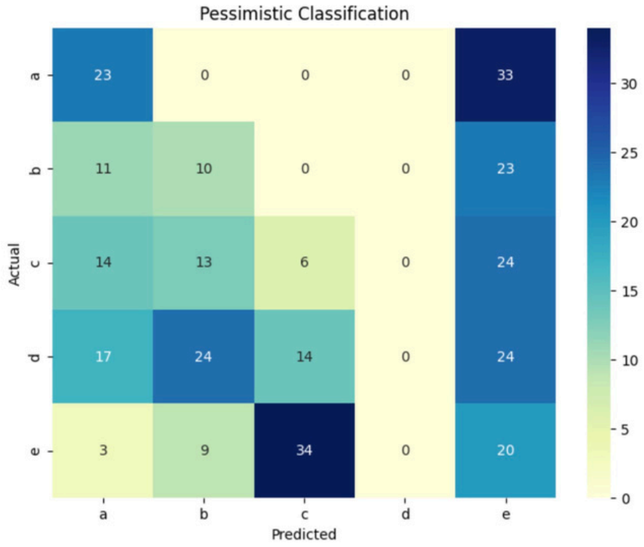
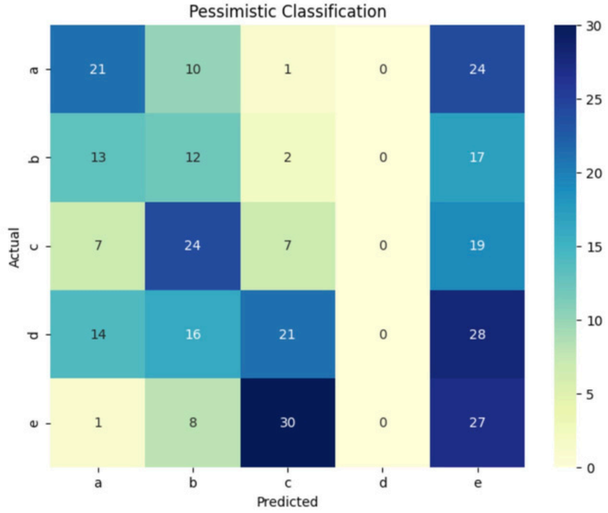
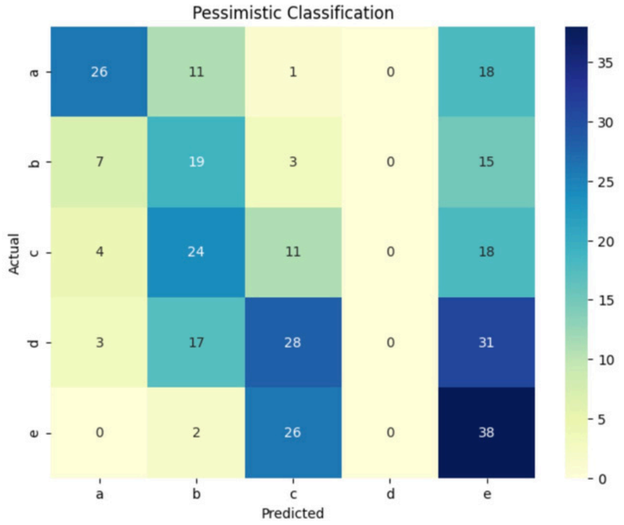
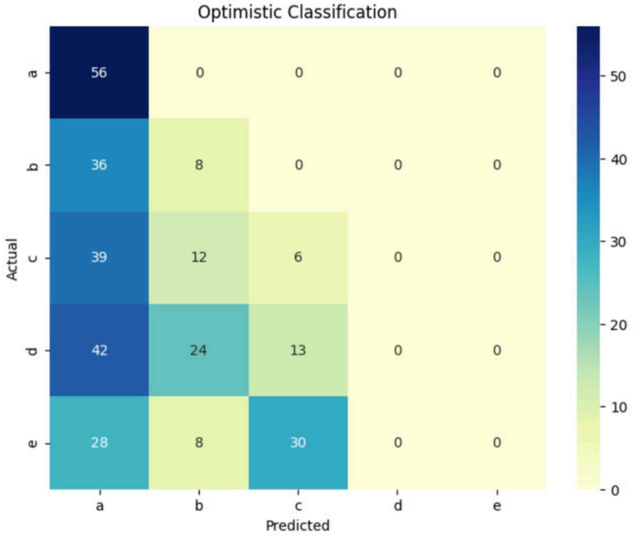
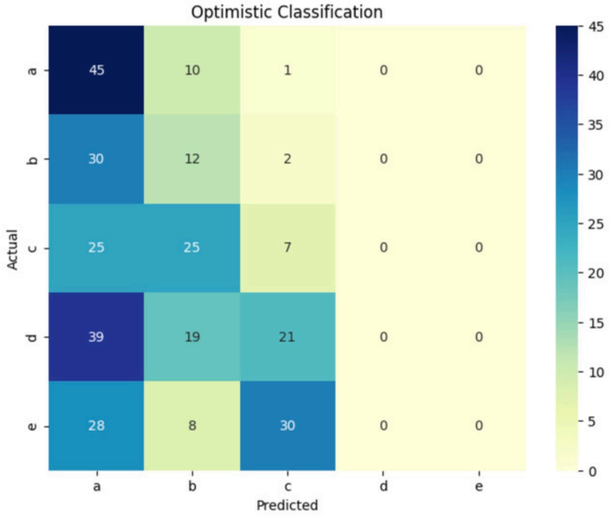
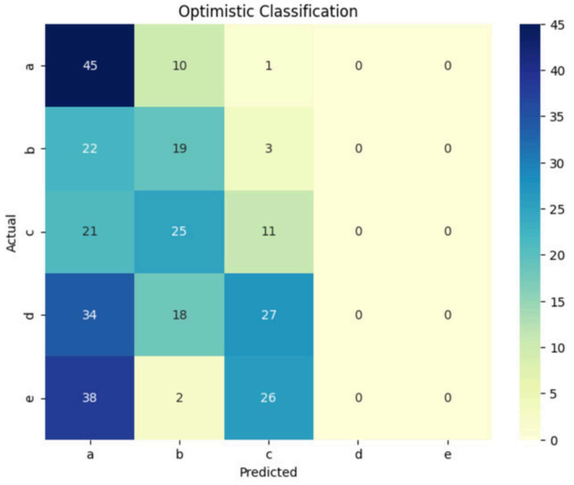




ELECTRE-Tri Model

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

Sorting Reults(Weights=4 3 3 3 2 2 1)

	λ = 0.5	λ = 0.6	λ = 0.7
Pessimistic			
Optimistic			

Our Findings

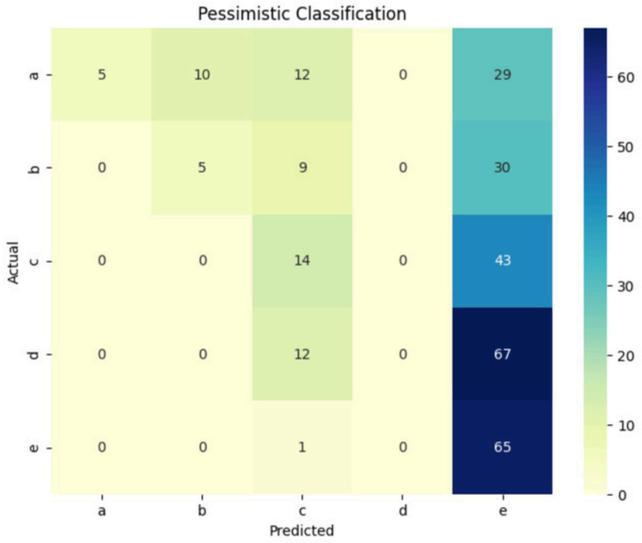
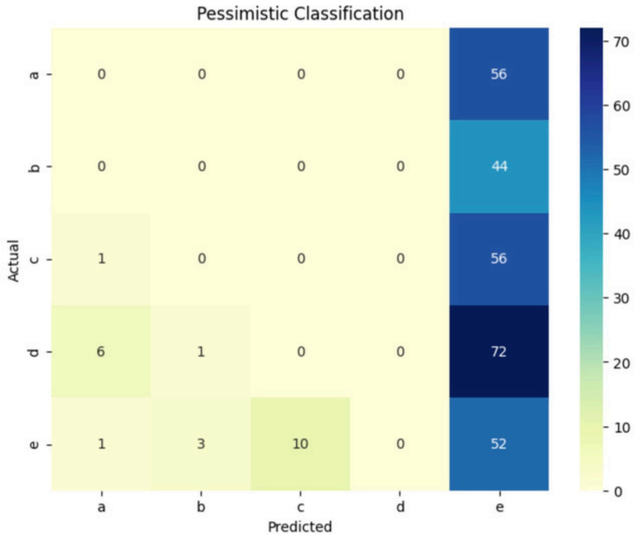
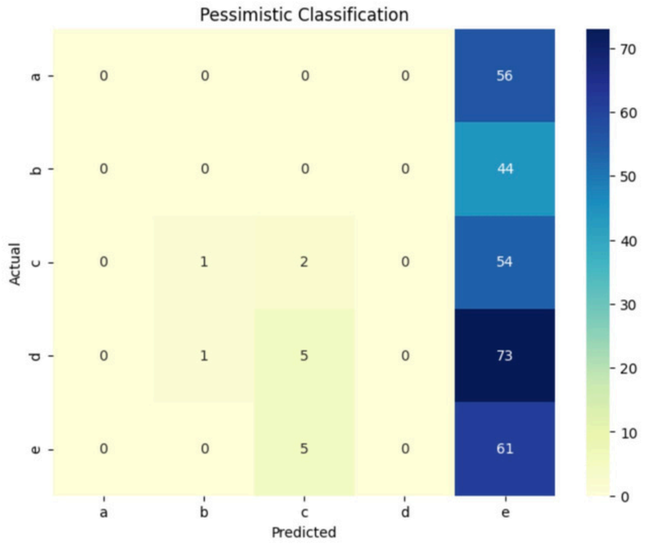
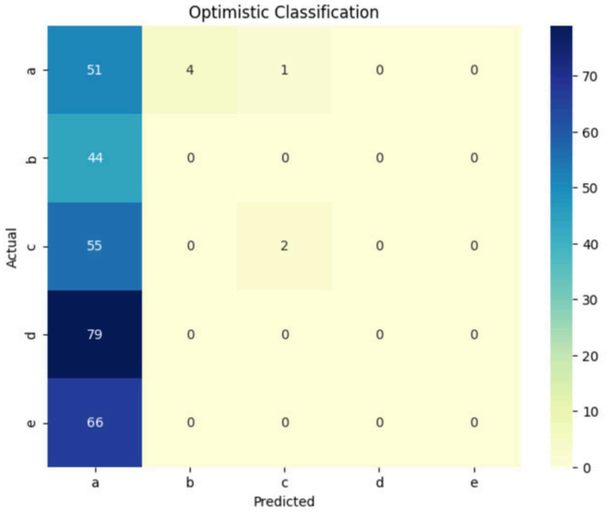
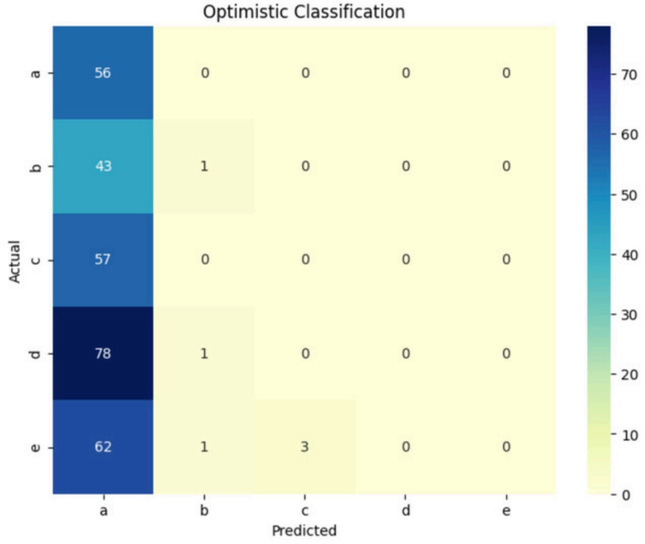
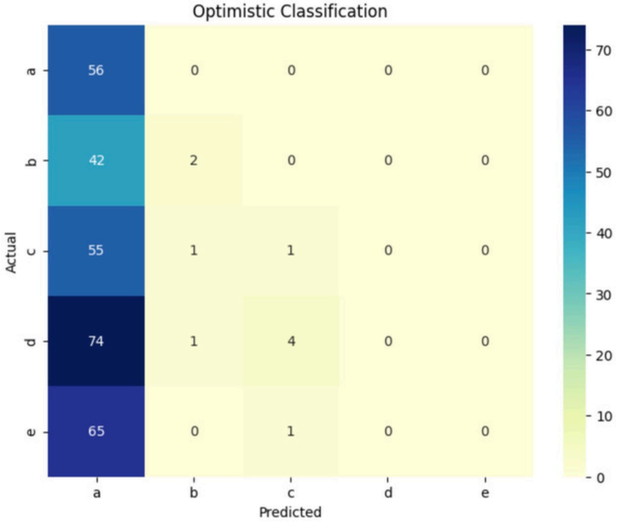
Optimistic vs. Pessimistic:

Optimistic Strategy: a > b > c > d > e (favors higher categories)
Pessimistic Strategy: a < b < c < d < e (favors lower categories)

Impact of λ on Classification Results:

λ = 0.5 (lenient): More samples in a and b categories
λ = 0.7 (strict): More samples in e categories

Sorting Reults(Weights=1 1 1 1 4 4 1)——more healthy

	λ = 0.5	λ = 0.6	λ = 0.7
Pessimistic			
Optimistic			

Our Findings

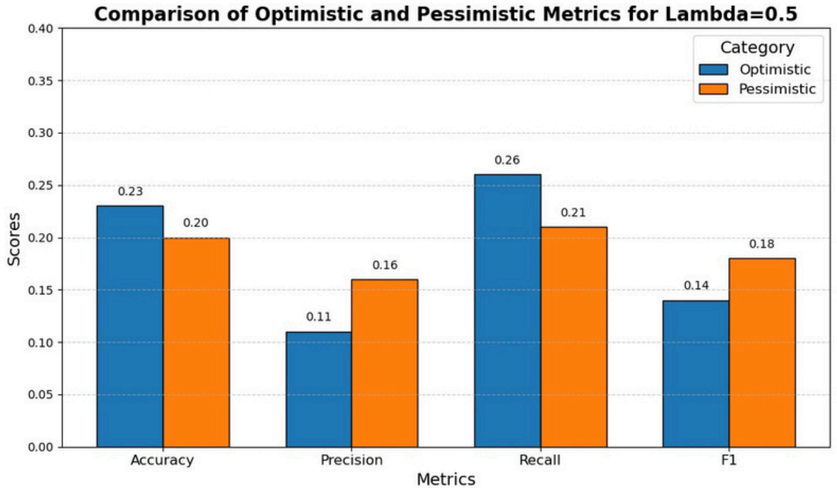
Optimistic vs. Pessimistic:
Optimistic Strategy: a > b > c > d > e (favors higher categories)
Pessimistic Strategy: a < b < c < d < e (favors lower categories)

Impact of λ on Classification Results:
λ = 0.5 (lenient): More samples in a and b categories
λ = 0.7 (strict): More samples in e categories

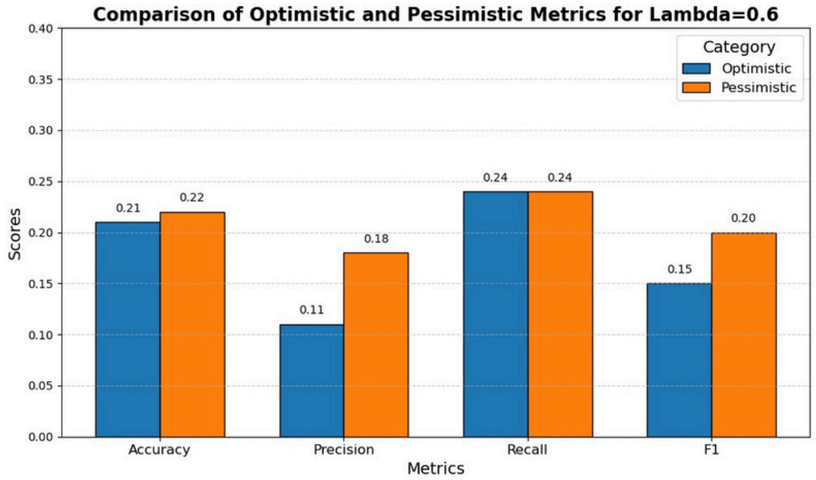
Impact of reducing the weight of unhealthy factors:
The number of “A” and “B” categories has increased, especially in the optimistic scenario.
Pessimistic classification strictly checks categories from highest to lowest, with high weights on healthy factors causing foods to be assigned to lower categories if they fail to meet healthy criteria.(more extreme)

Comparison of Optimistic and Pessimistic Metrics

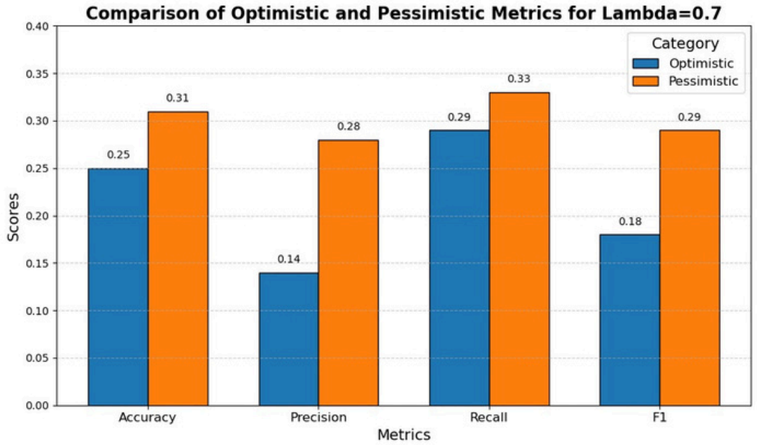
λ = 0.5



λ = 0.6



λ = 0.7



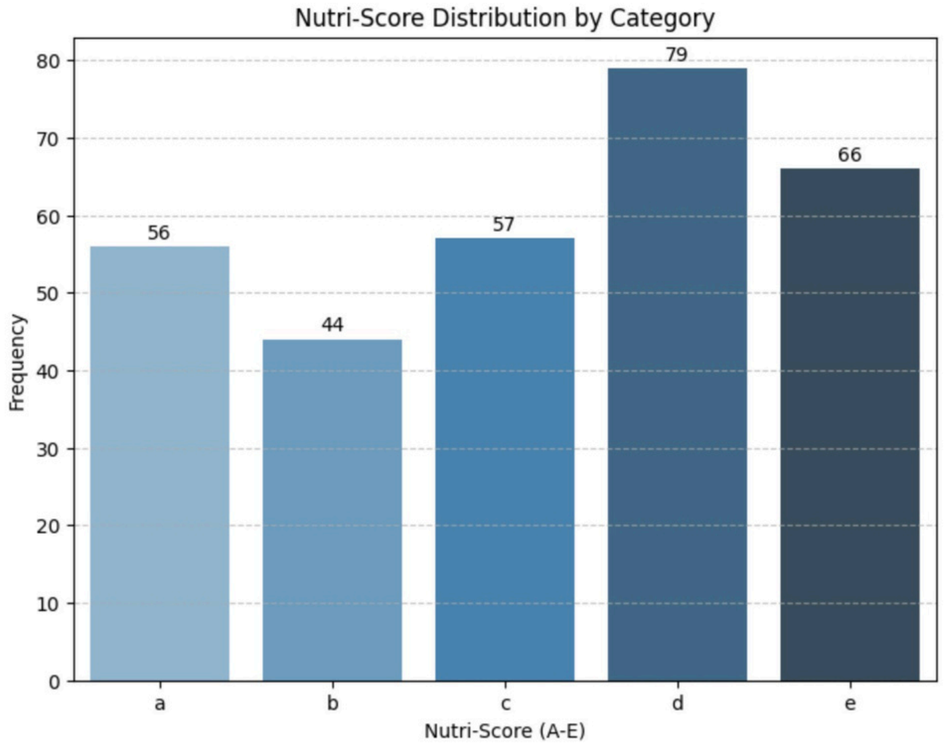
As λ increased, the Pessimistic model showed notable improvements across all metrics, benefiting from stricter thresholds that reduced false positives and enhanced consistency. Conversely, the Optimistic model exhibited limited gains due to its lenient strategy, leading to weaker precision-recall balance.

Higher λ thresholds led to stricter categorization, improving both models' performance, though the Optimistic model sacrificed recall for precision. The Pessimistic model consistently outperformed the Optimistic model, especially at higher thresholds, prioritizing precision and accuracy. The data's inherent imbalance (e.g., more samples in lower-tier categories like "D" or "E") likely favored the Pessimistic model's focus on majority class accuracy.

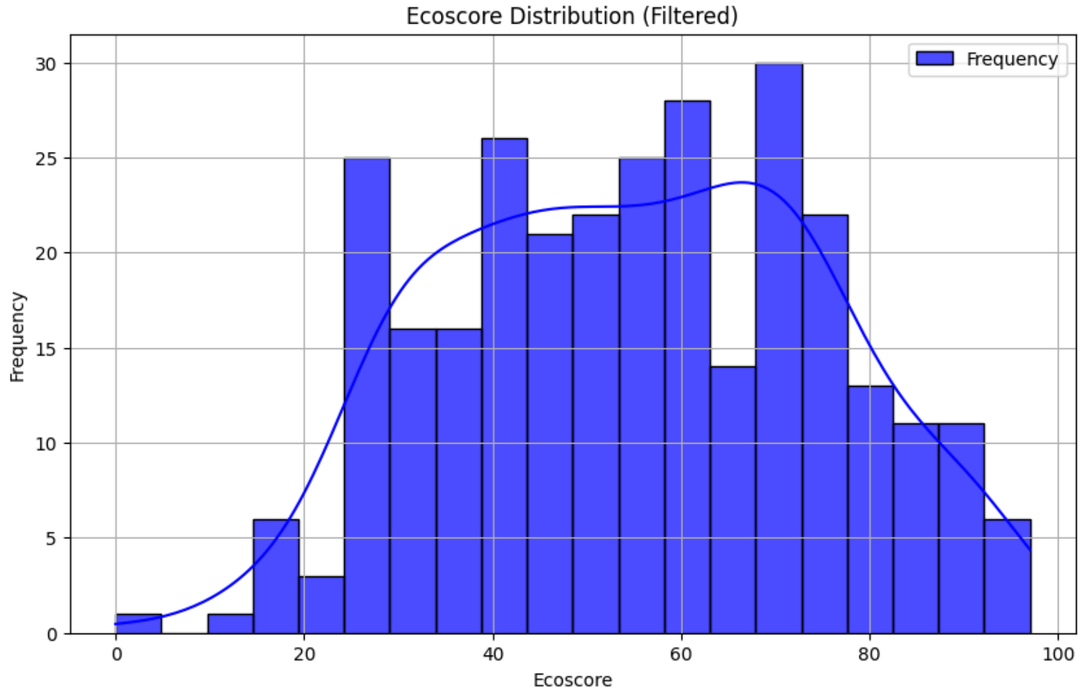
03

Combine Nutri Eco Score using two MCDA Models

Distributions of two Scores



Nutri-Score:from e to a.



Eco-Score:from 0 to 100 (from e to a)

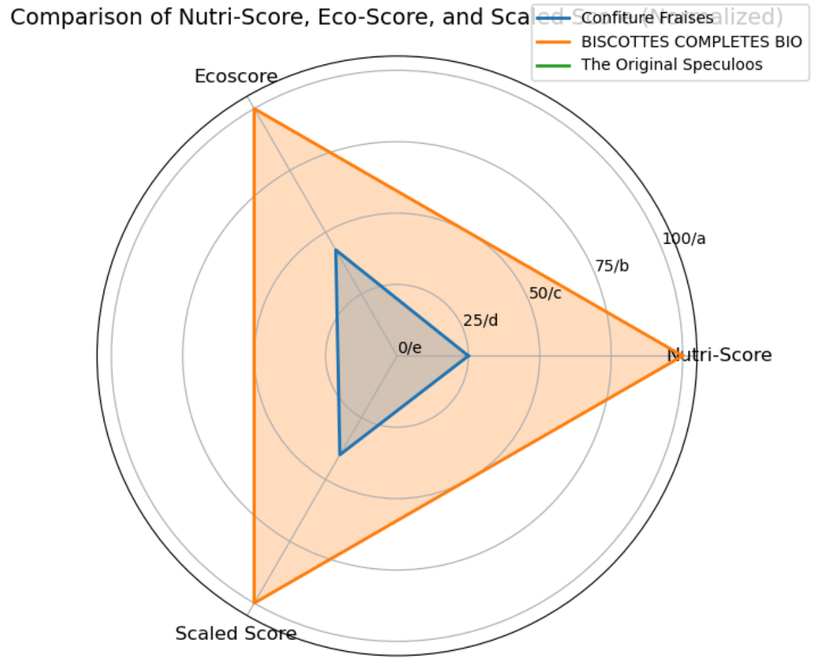
fixed threshold values to assign categories (20, 40, 60, 80)

Weighted Sum Model

1.Min-Max Normalization: $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$

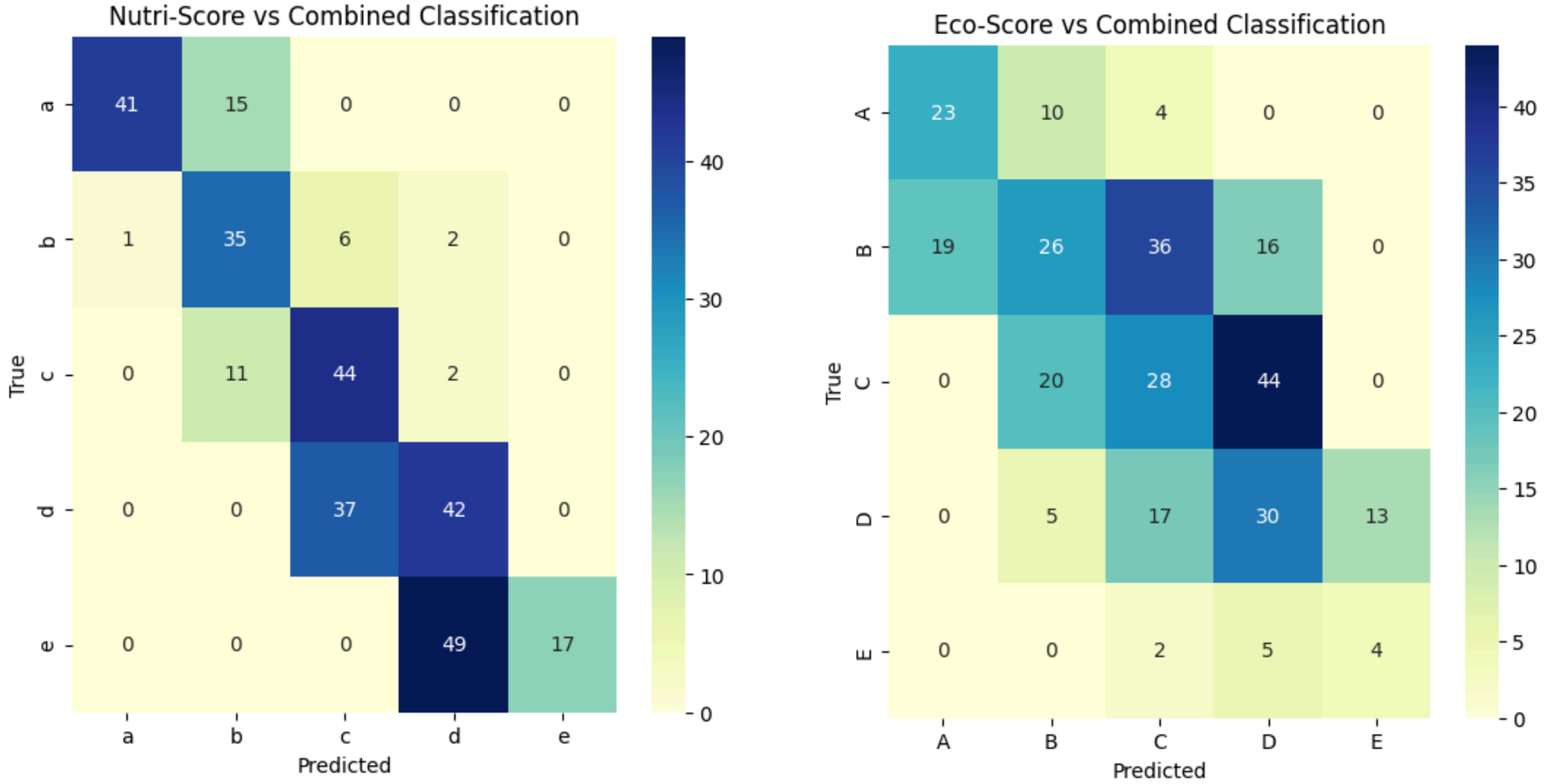
For Nutri-Score:a=5,b=4,c=3,d=2,e=1
Then : N1 = (Nutri_Score - 1) / (5-1)

For Eco-Score(from 0 to 100):
Then : E1=(Eco-Score-0) / (100-0)



2.**Weight** set w1=50%,w2=50%
3.**Sum**=w1*N1+w2*E1
4.**Sorting**: A(0.8~1),B(0.6~0.8),C(0.4~0.6),D(0.2~0.4),E(0~0.2)

Sorting Reults Comparison



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

Our Findings

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

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

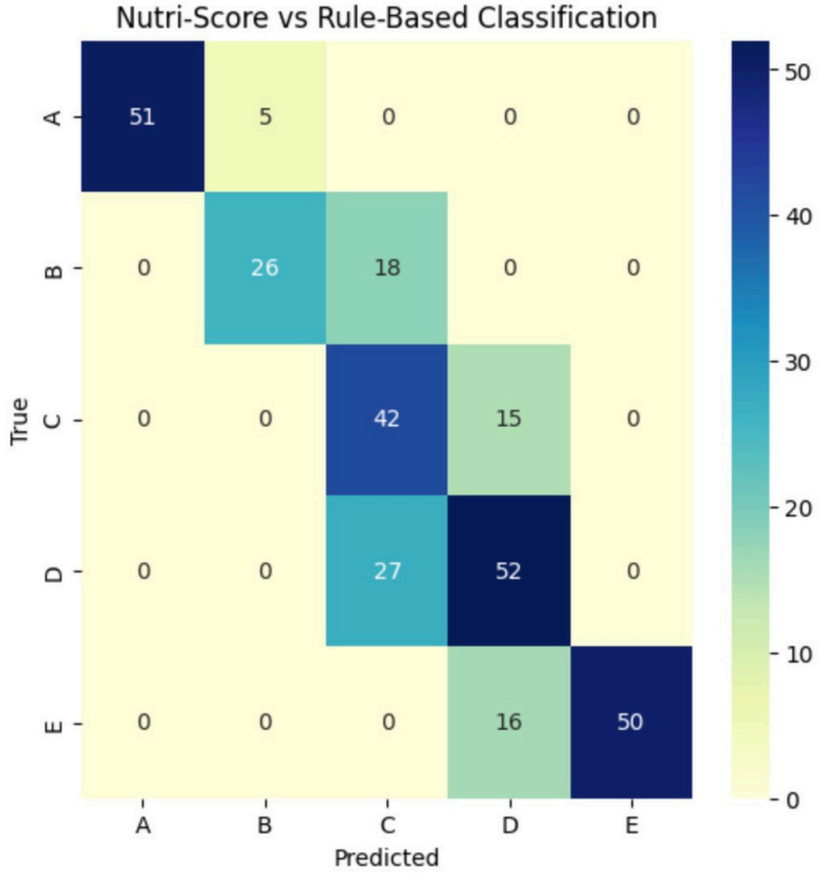
Threshold-based classification ensures consistency:
The classification thresholds for Nutri-Score and Eco-Score are carefully designed. If these thresholds align well with the distribution of the combined scores, the classifications derived from the weighted sum will naturally align with the original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

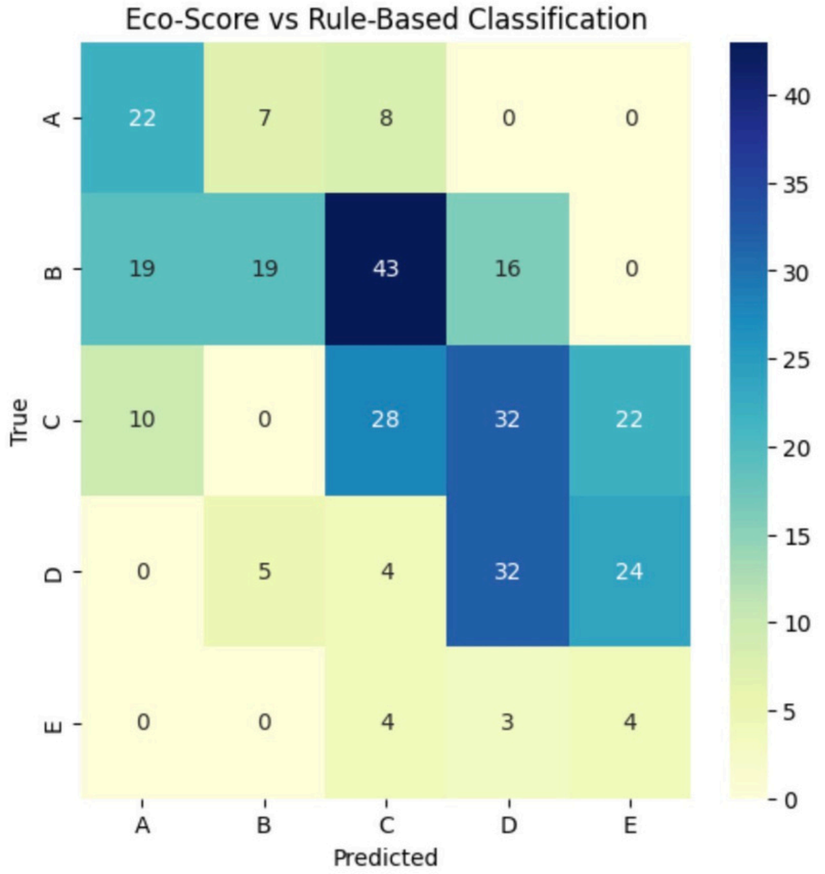
Simple Decision Rules Model

We set the rules to prioritize the **primary dimension (Nutri-Score**) while using the **secondary dimension (Eco-Score**) for adjustments when appropriate.

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

Sorting Reults Comparison





The results of the simple decision rule model is more consistent with the original Nutri-Score.

Our Findings

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

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

2. Limited Influence of Eco-Score

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

3. Aligned Categorization Frameworks

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

04

Machine Learning Models

Results and Confusion Matrices

Machine Learning Models

Random Forest

Setup:

Max Depth: Automatically determined.

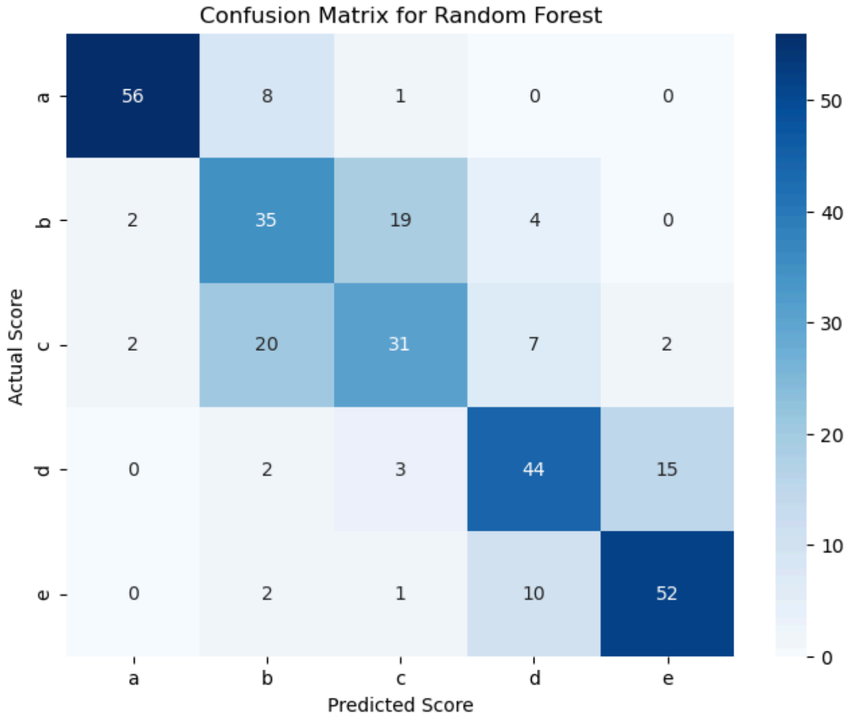
Insights:

Accuracy: 68.99%.

The model performs relatively well in predicting all classes.

Class predictions show a balanced distribution across all categories.

Shows minor bias toward frequent classes.



K-Nearest Neighbors (KNN)

Setup:

Number of Neighbors: 5.

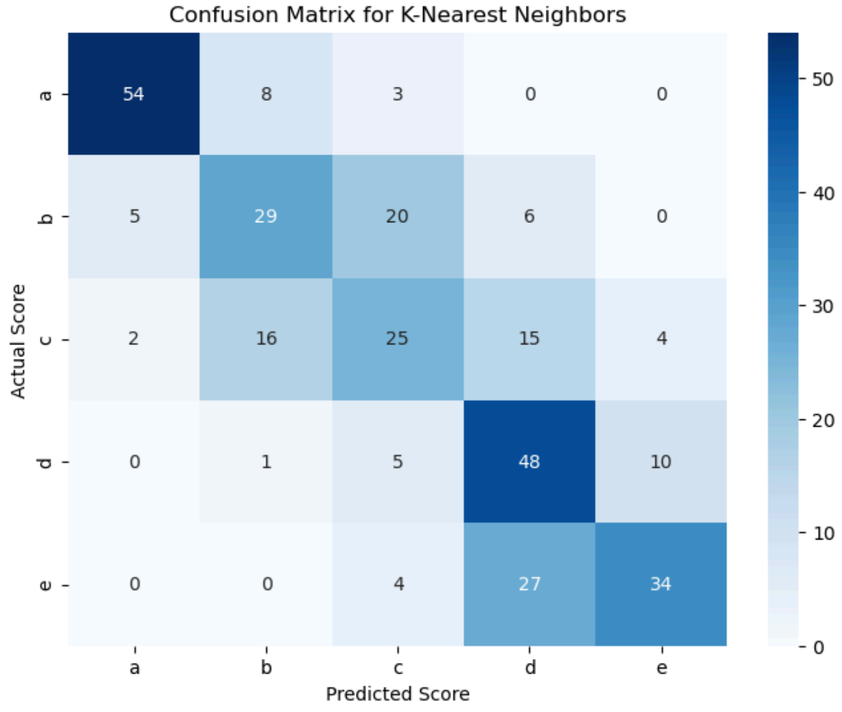
Insights:

Accuracy: 60.13%.

Performs moderately, with a tendency to confuse neighboring classes.

Struggles to differentiate lower-probability classes.

Misclassifications often occur between adjacent classes.



Machine Learning Models

Gaussian Naive Bayes

Setup:

Based on Gaussian distribution of features.

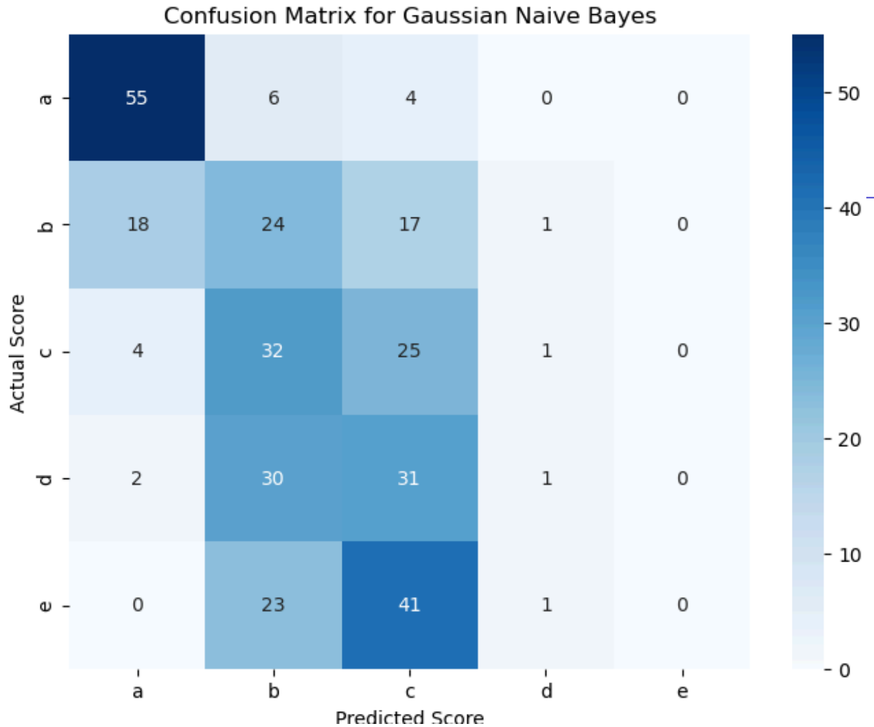
Insights:

Accuracy: 33.23%.

Struggles significantly with imbalanced data.

Assumption about features are non-related is incorrect.

Confusion matrix reveals substantial bias toward specific classes.



XGBoost

Setup:

Booster: GBTree.

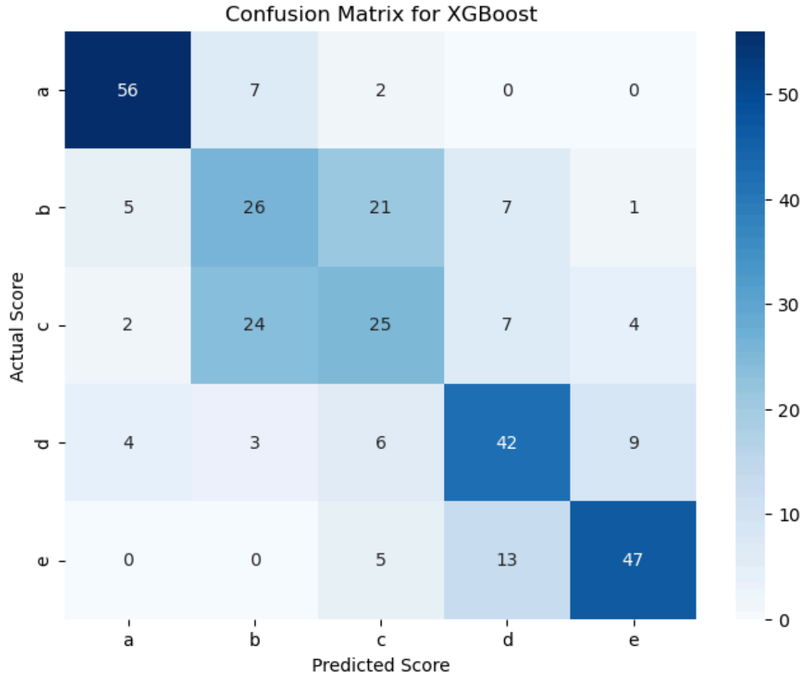
Insights:

Accuracy: 62.03%.

Performance is slightly better than KNN but lower than Random Forest.

Boosting method perform worse in small dataset.

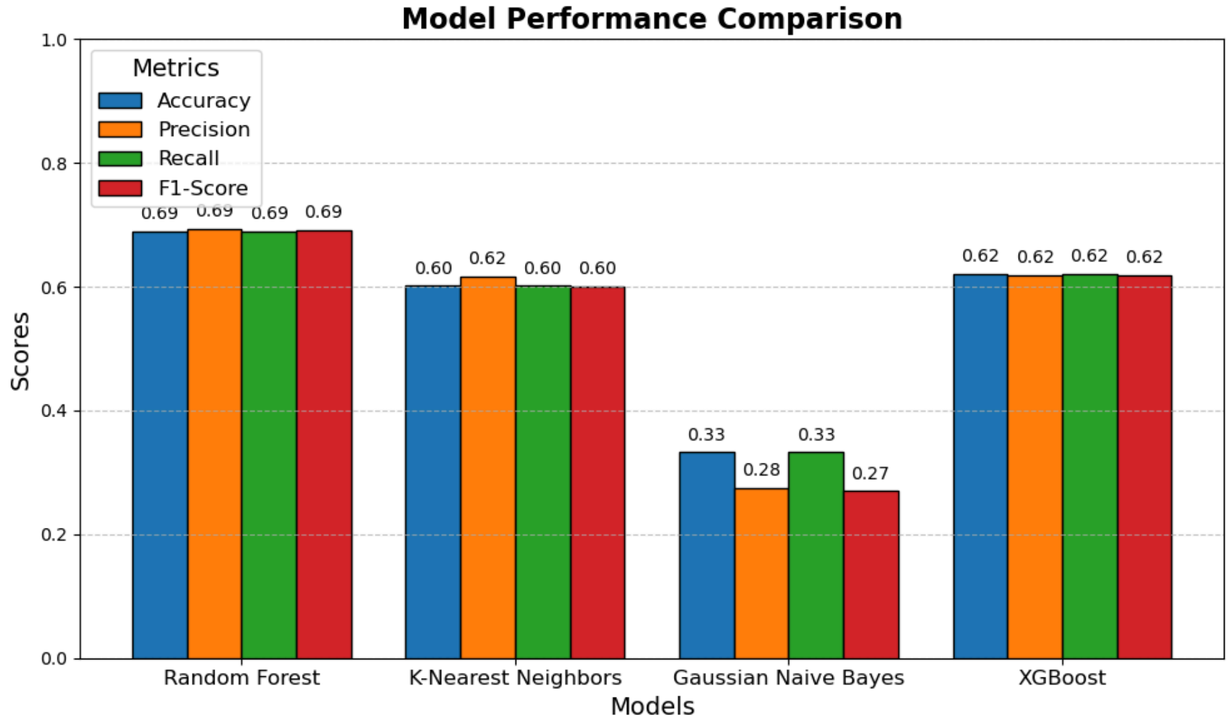
Shows better recall for intermediate classes compared to Naive Bayes.



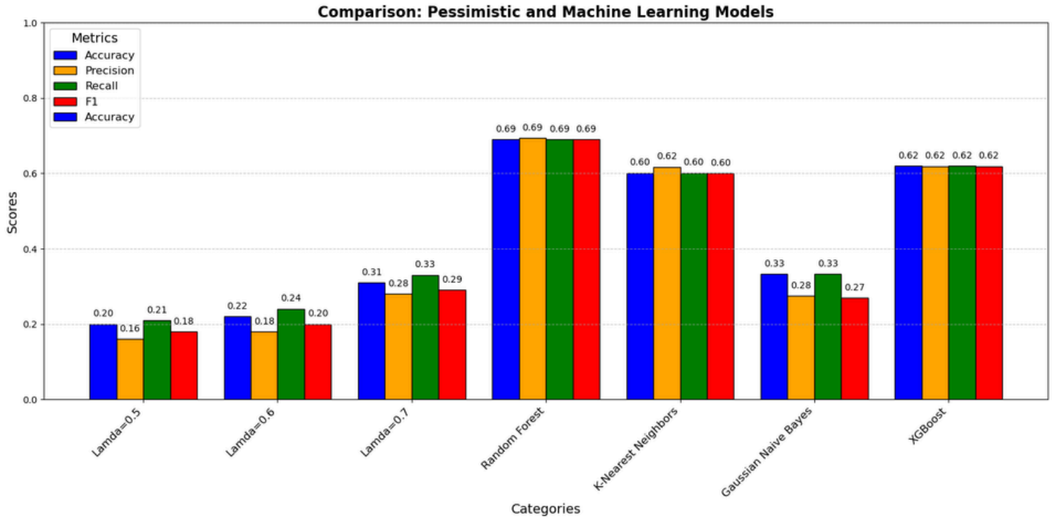
Machine Learning Models

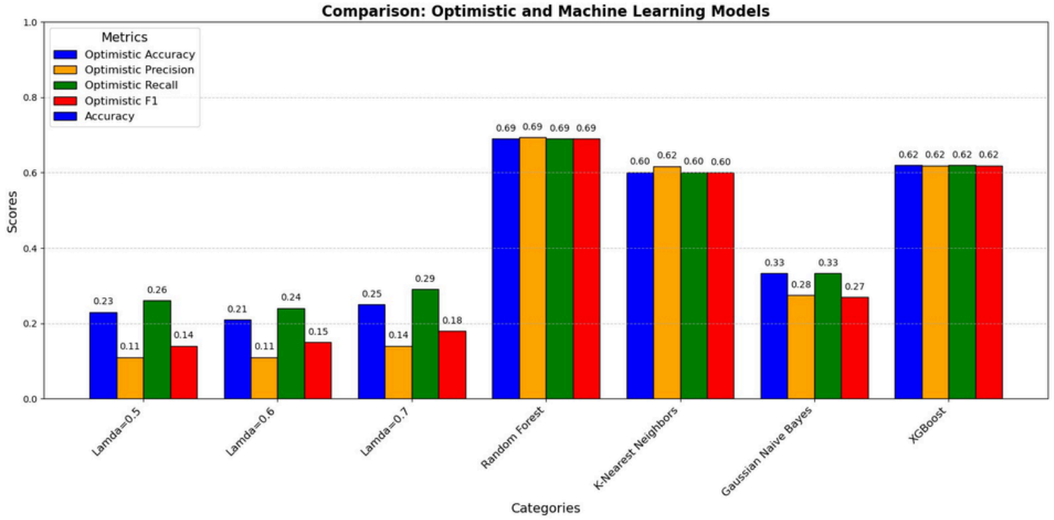
Summary

- Train-Test Split: An 80:20 split was used
- Class Imbalance: Uneven distributions of classes, impacting performance metrics across all models.
- Lack of Data: All models could benefit significantly from additional unbiased data to improve both true positive rates and reduce false negatives.



Comparison Between ELECTRE-Tri Model and Machine Learning Models





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

Machine learning models outperformed Optimistic Classification across all metrics, with Random Forest achieving the highest Accuracy (0.69) and F1-Score (0.69) compared to Optimistic's best Accuracy (0.25) and F1-Score (0.18).

Optimistic Classification prioritized Recall (0.29) but suffered from low Precision (0.14) and overall poor Accuracy.

Pessimistic Classification performed better than Optimistic but lagged behind machine learning models.

At its best (λ=0.7λ=0.7), Pessimistic achieved Accuracy (0.31) and F1-Score (0.29), much lower than models like Random Forest or XGBoost.

Its conservative nature improved Precision (0.28) but sacrificed Recall compared to machine learning models.

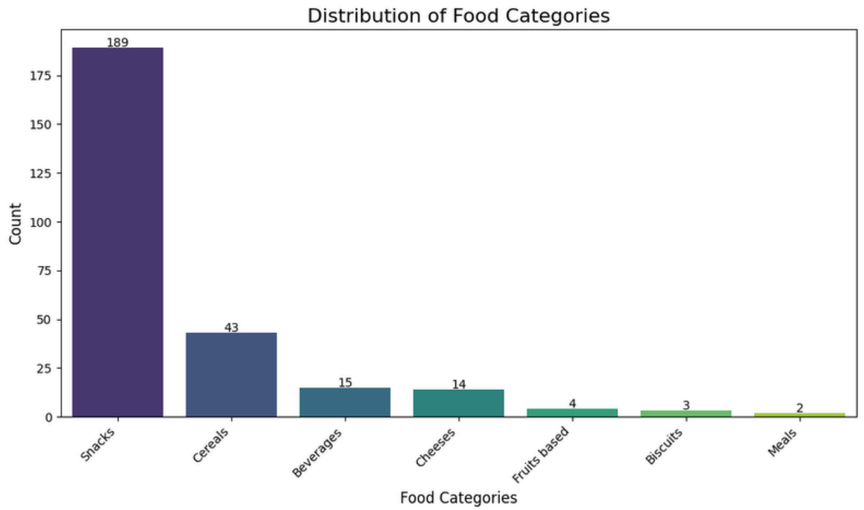
05

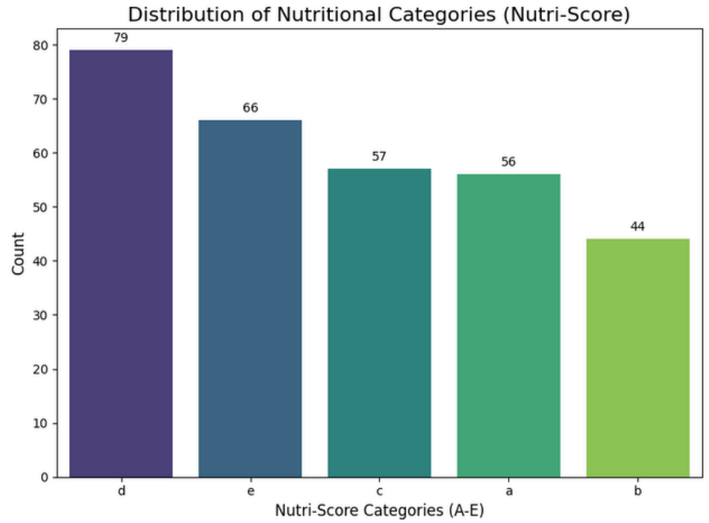
Comparison

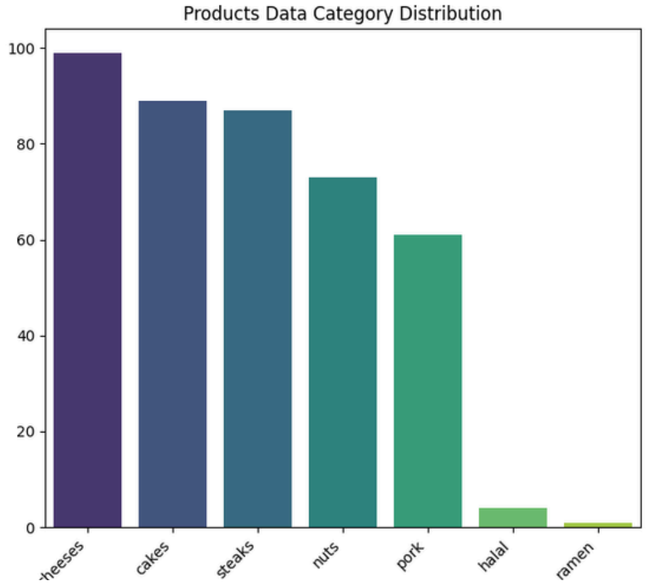
Compare our MCDA model with another group!

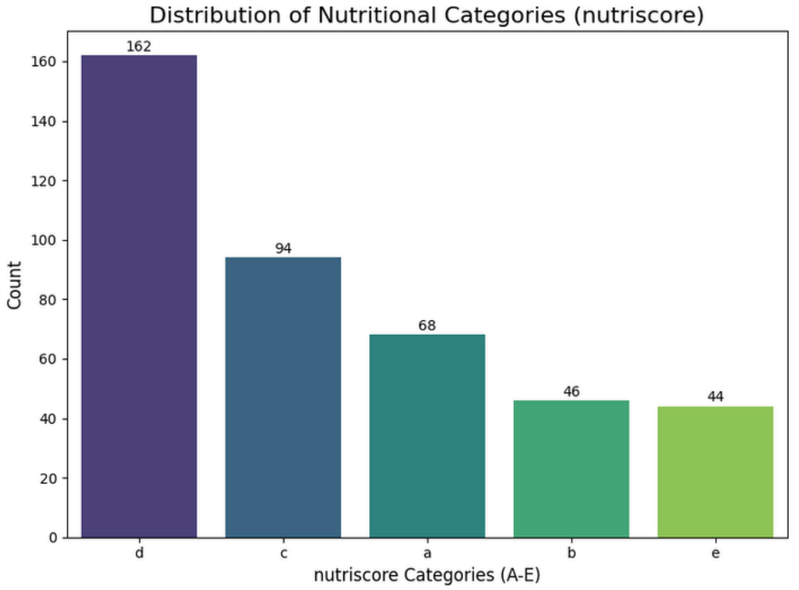
Comparison(with group of Gabriel,Qasim, Ben)

Our data distribution compare to Beard Guys' data



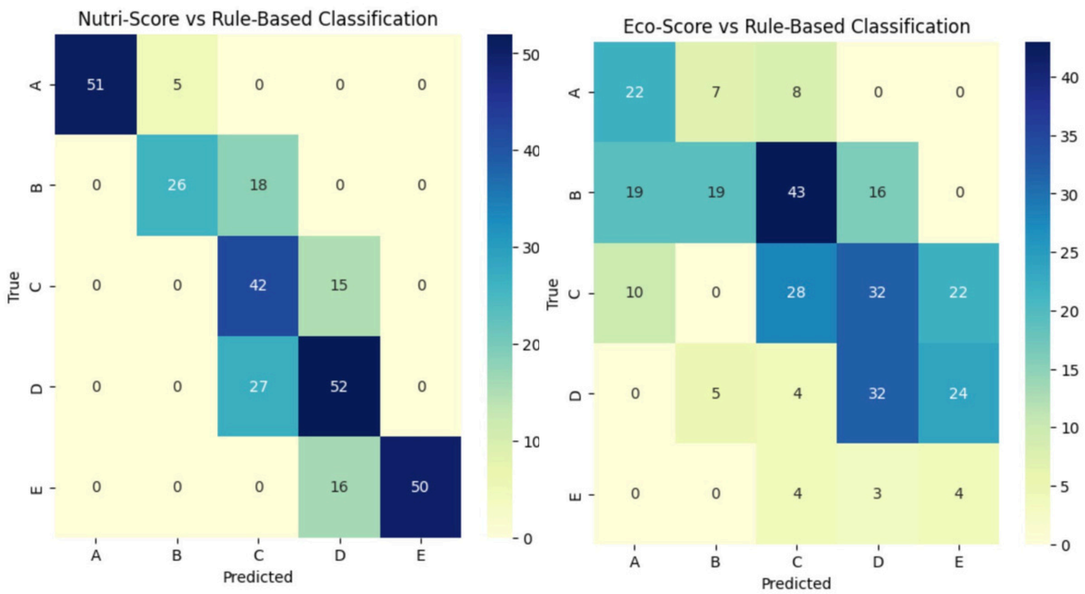


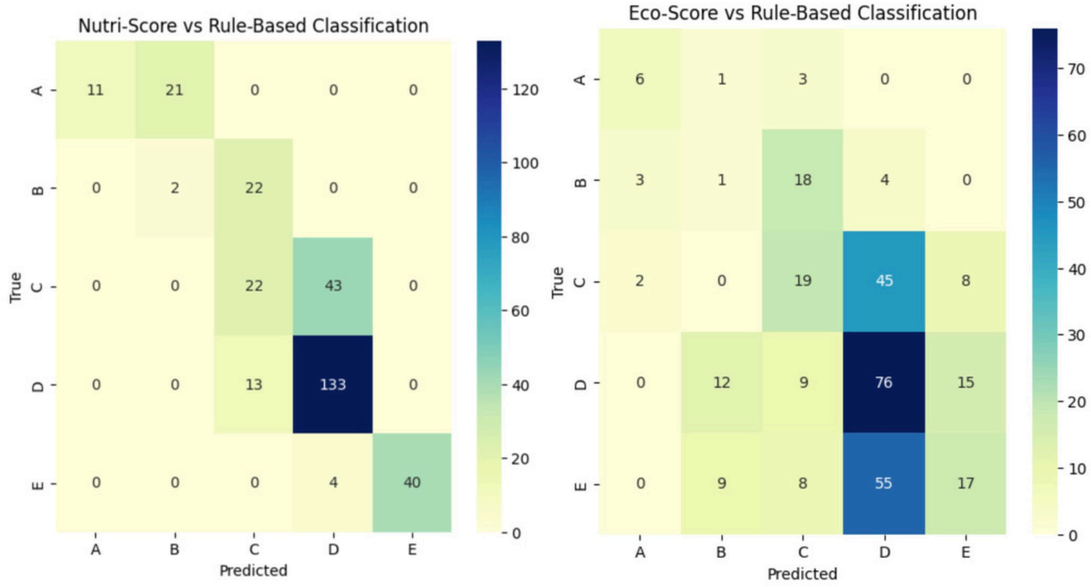




Comparison

Our MCDA Model Simple Decision Rules compared to our data & Beard Guys' data





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

Conclusion about models

Most Comfortable Model

The **Random Forest model** was the most comfortable, with high performance across all metrics (Accuracy: 0.69, F1-Score: 0.69) and robust handling of imbalanced data.

Most Suitable Model

The Pessimistic ELECTRE-Tri model (λ=0.7) was best suited for calculating Nutri-Score due to its simplicity, alignment with predefined thresholds, and reasonable performance (Accuracy: 0.31, F1-Score: 0.29).

Most Explainable Model

The **Pessimistic ELECTRE-Tri model (λ=0.7)** stood out for its transparency and ease of interpretation.

THANKS

Supervisor:
Prof. Brice MAYAG

Presented by:
Xianyun Zhuang
Jintao Ma
Yutao Chen